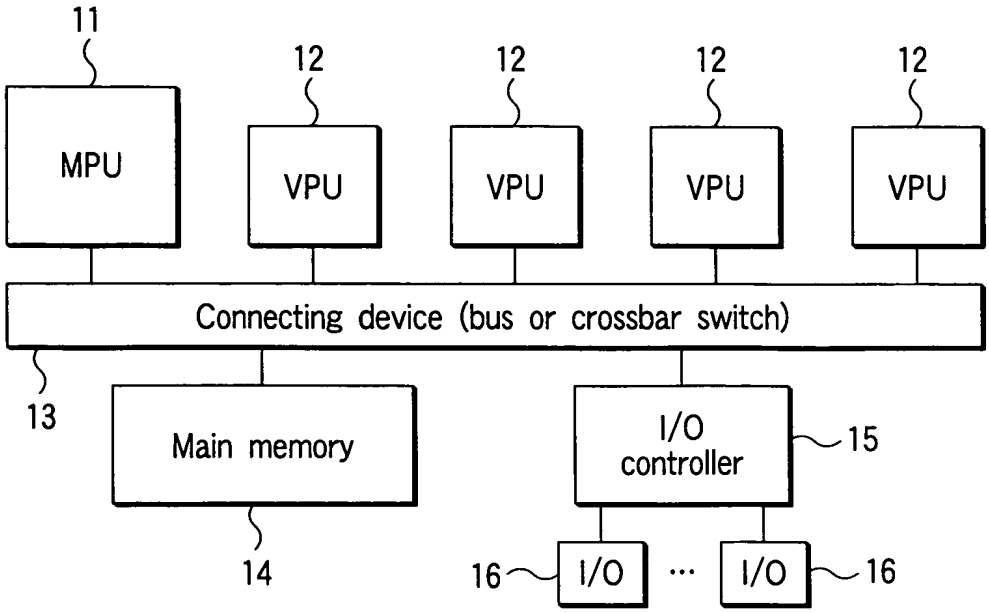
11
12
12
12
12
MPU
VPU
VPU
VPU
VPU
Connecting device (bus or crossbar switch)
13
Main memory
I/O controller
15
14
16
I/O
I/O
16

FIG. 1

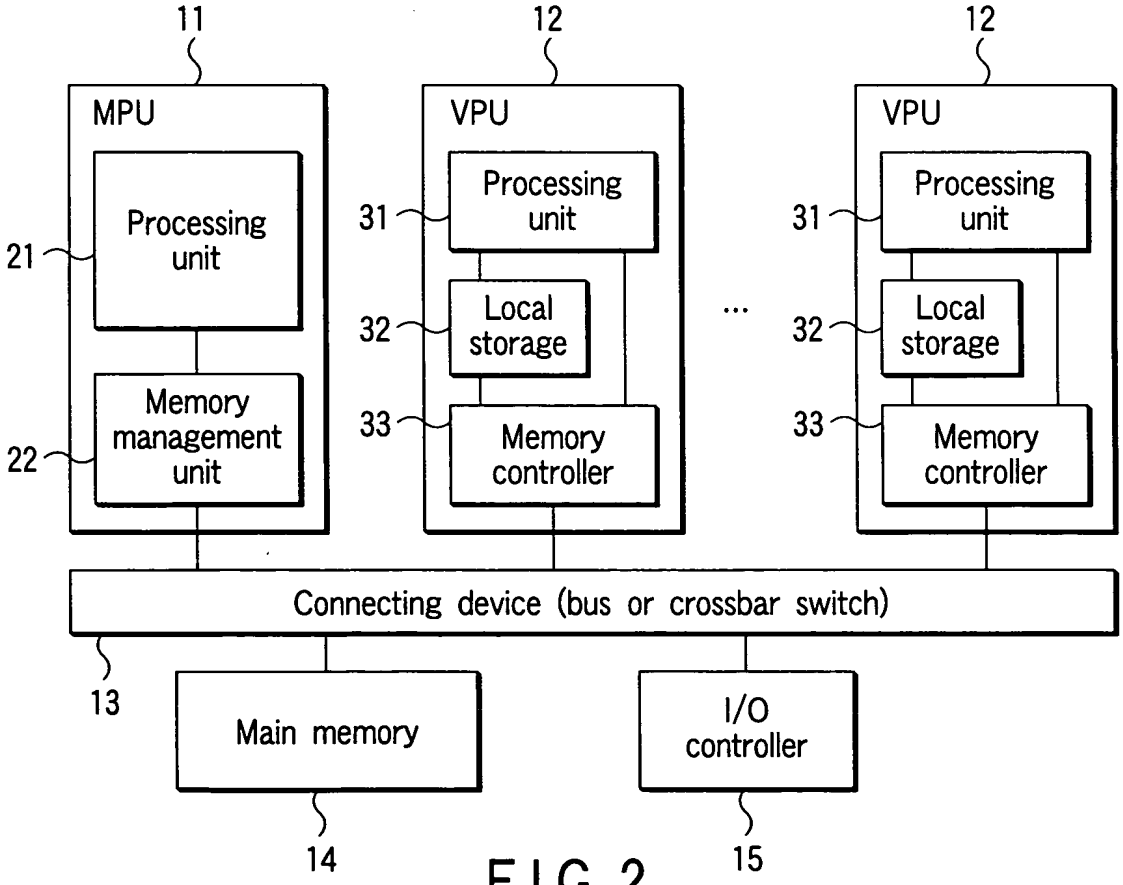
11
12
12
MPU
VPU
VPU
Processing unit
21
Memory management unit
22
31
Processing unit
32
Local storage
33
Memory controller
31
Processing unit
32
Local storage
33
Memory controller
Connecting device (bus or crossbar switch)
13
Main memory
I/O controller
14
15

FIG. 2

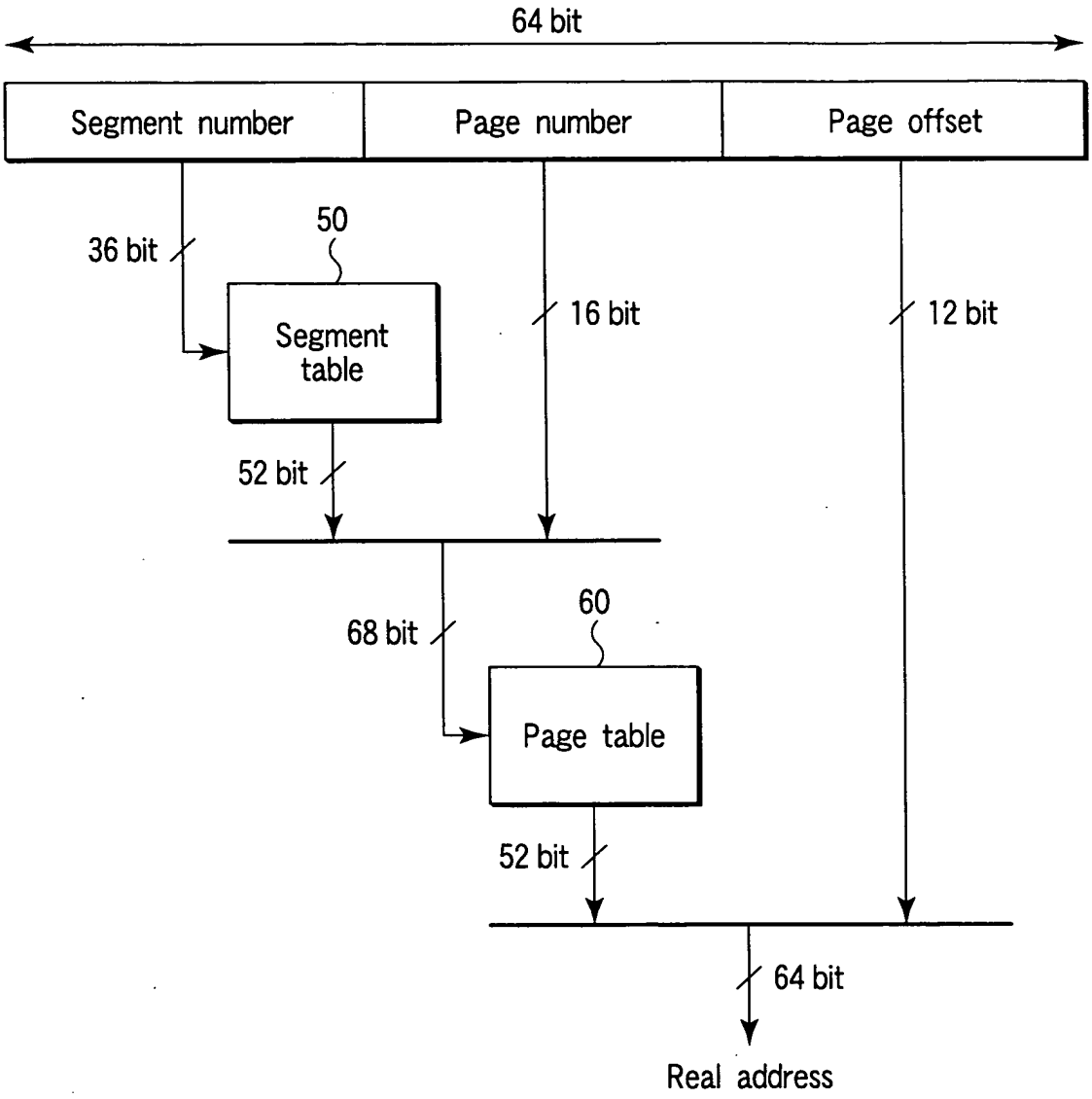
64 bit
Segment number
Page number
Page offset
36 bit
50
Segment table
16 bit
12 bit
52 bit
68 bit
60
Page table
52 bit
64 bit
Real address

FIG. 3

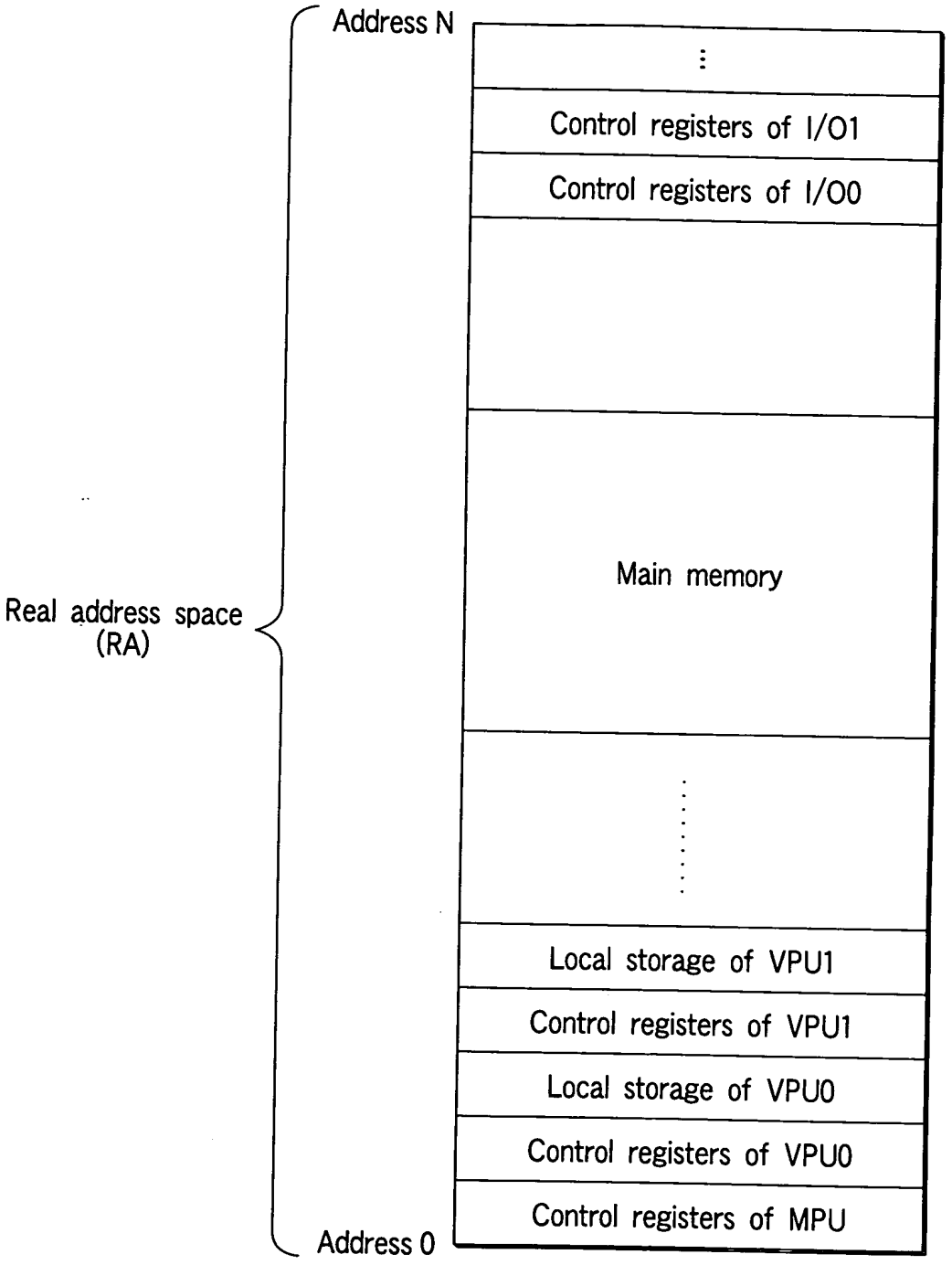
Address N
Control registers of I/O1
Control registers of I/O0
Main memory
Local storage of VPU1
Control registers of VPU1
Local storage of VPU0
Control registers of VPU0
Control registers of MPU
Address 0
Real address space
(RA)

FIG. 4

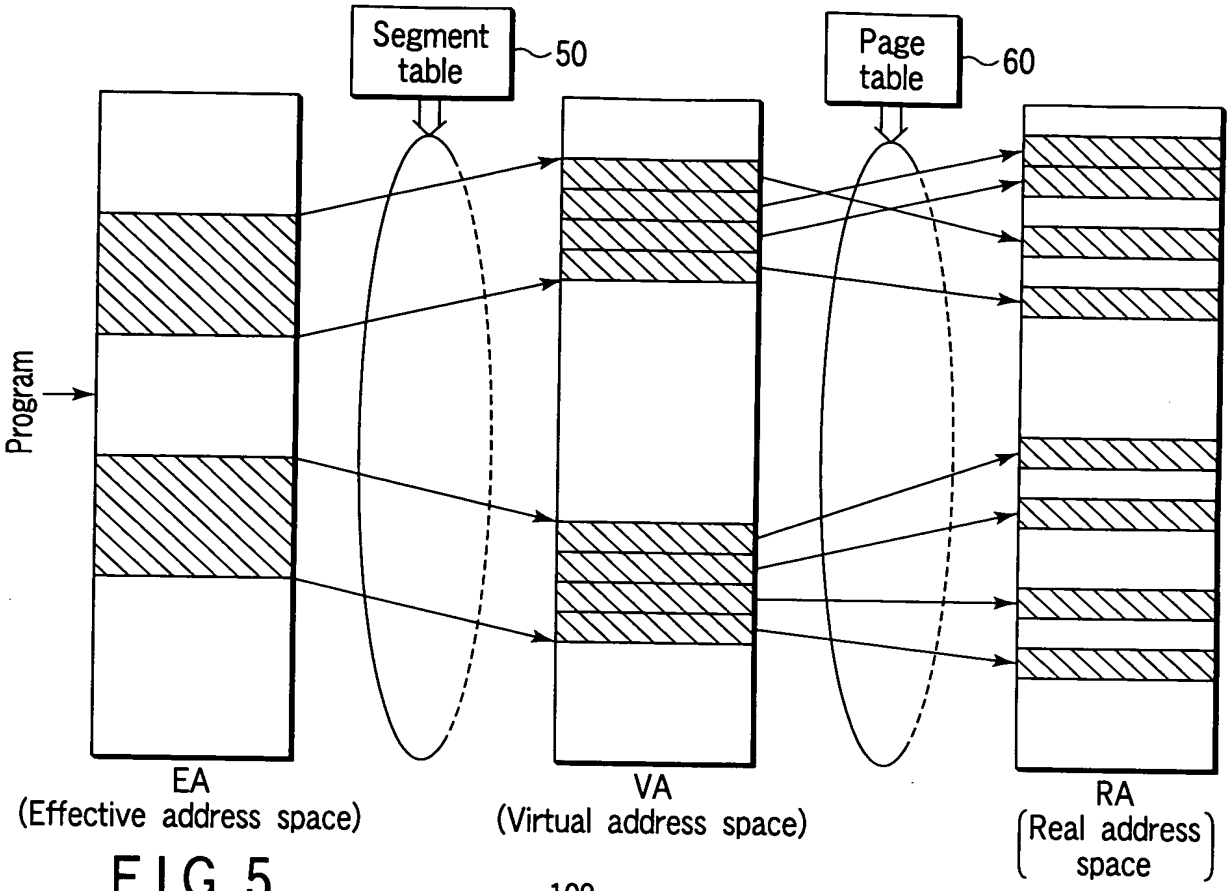


FIG. 5

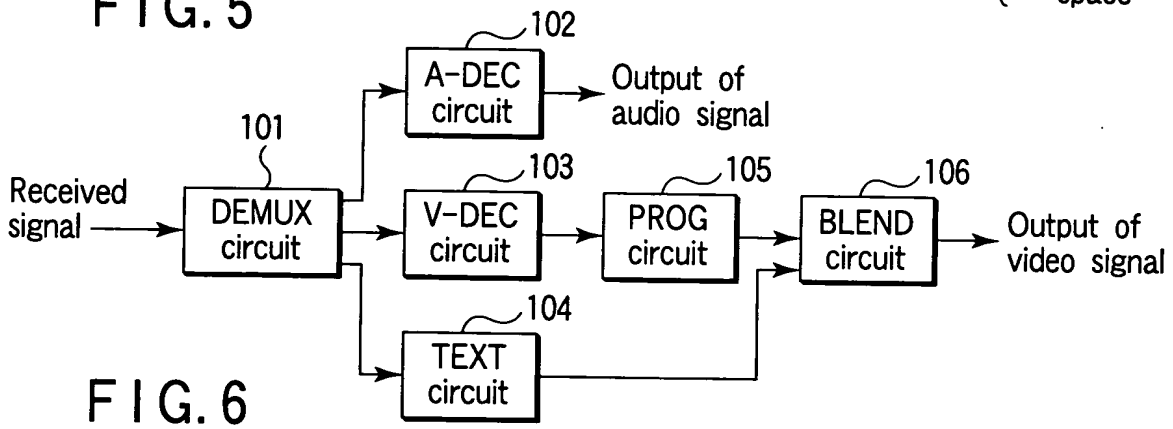


FIG. 6

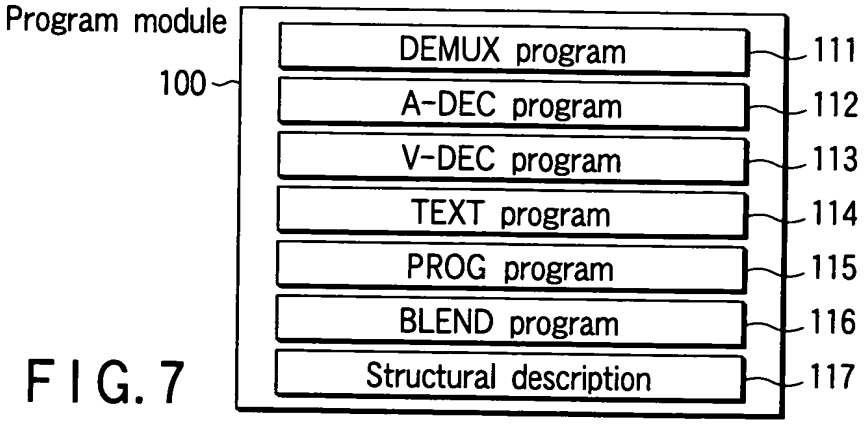


FIG. 7

Structural description 117

| Number | Program | Input | Output | Cost | Buffer |
|---|---|---|---|---|---|
| (1) | DEMUX | Received signal | (2)<br>(3)<br>(4) | 5 | 100KB<br>1MB<br>10KB |
| (2) | A-DEC | (1) | Audio output | 10 | ——— |
| (3) | V-DEC | (1) | (5) | 50 | 1MB |
| (4) | TEXT | (1) | (6) | 5 | 10KB |
| (5) | PROG | (3) | (6) | 20 | 1MB |
| (6) | BLEND | (4)<br>(5) | Video output | 10 | ——— |

Thread parameters

( • Tightly coupled thread group :
• Loosely coupled thread group : )

Others

FIG. 8

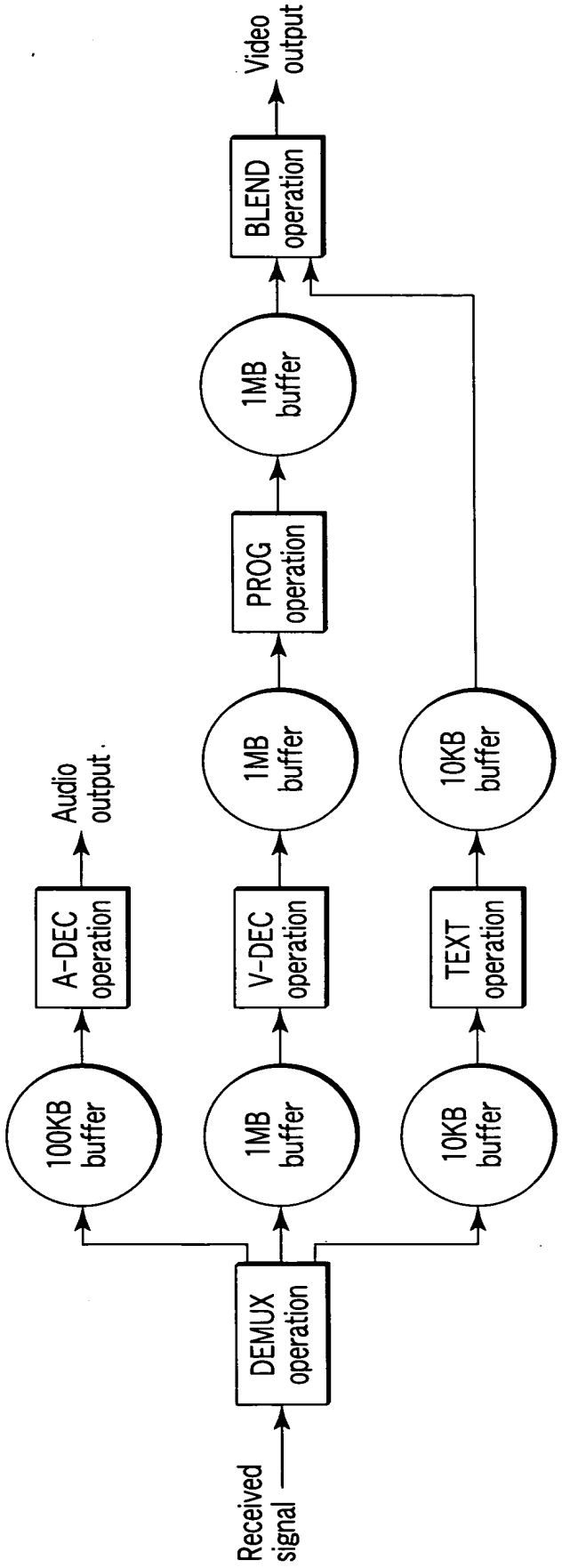
Received signal
DEMUX operation
100KB buffer
1MB buffer
10KB buffer
A-DEC operation
V-DEC operation
TEXT operation
Audio output
1MB buffer
10KB buffer
PROG operation
1MB buffer
BLEND operation
Video output

FIG. 9

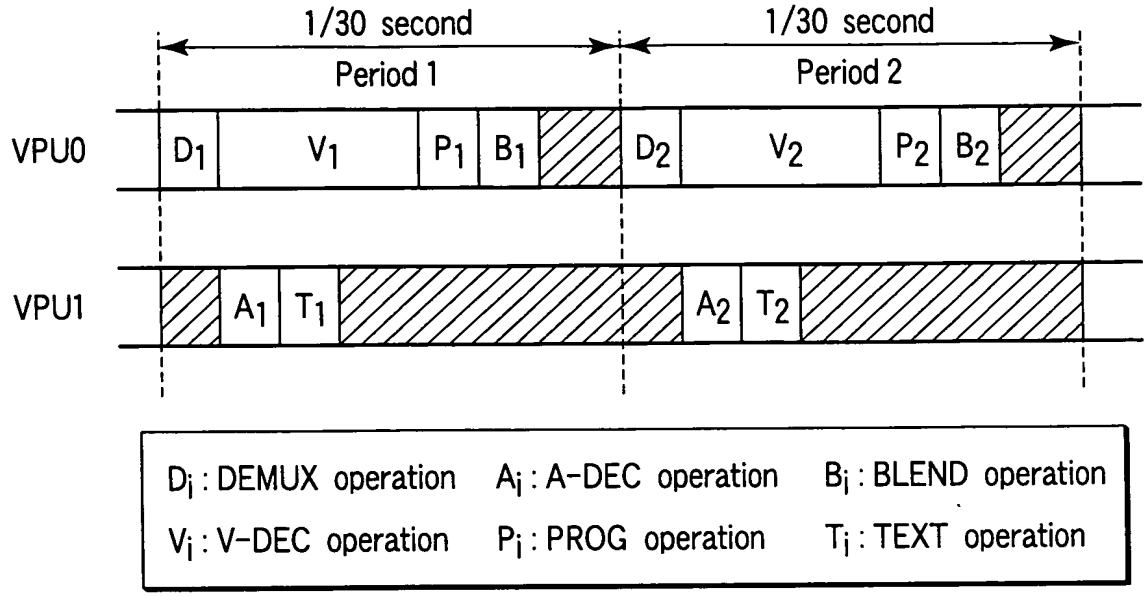
1/30 second
1/30 second
Period 1
Period 2
VPU0
D1
V1
P1
B1
D2
V2
P2
B2
VPU1
A1
T1
A2
T2
Di : DEMUX operation
Ai : A-DEC operation
Bi : BLEND operation
Vi : V-DEC operation
Pi : PROG operation
Ti : TEXT operation

FIG. 10

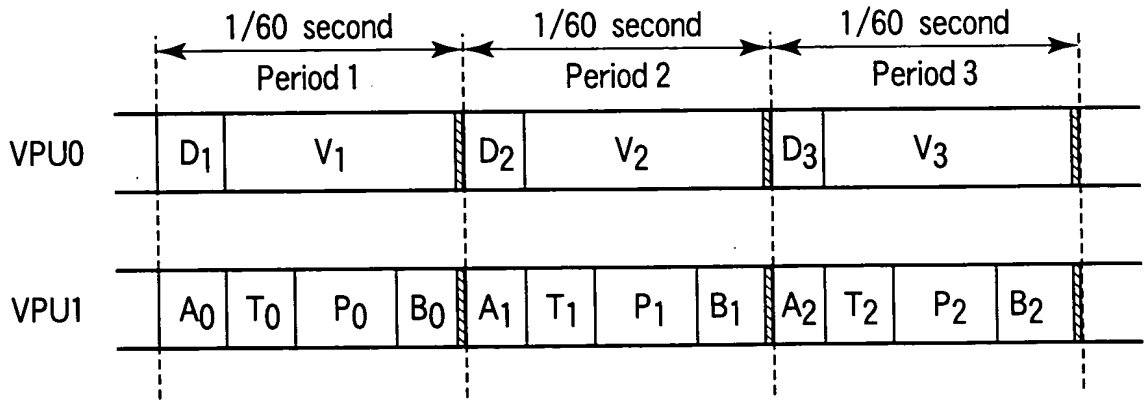
1/60 second
1/60 second
1/60 second
Period 1
Period 2
Period 3
VPU0
D1
V1
D2
V2
D3
V3
VPU1
A0
T0
P0
B0
A1
T1
P1
B1
A2
T2
P2
B2

FIG. 11

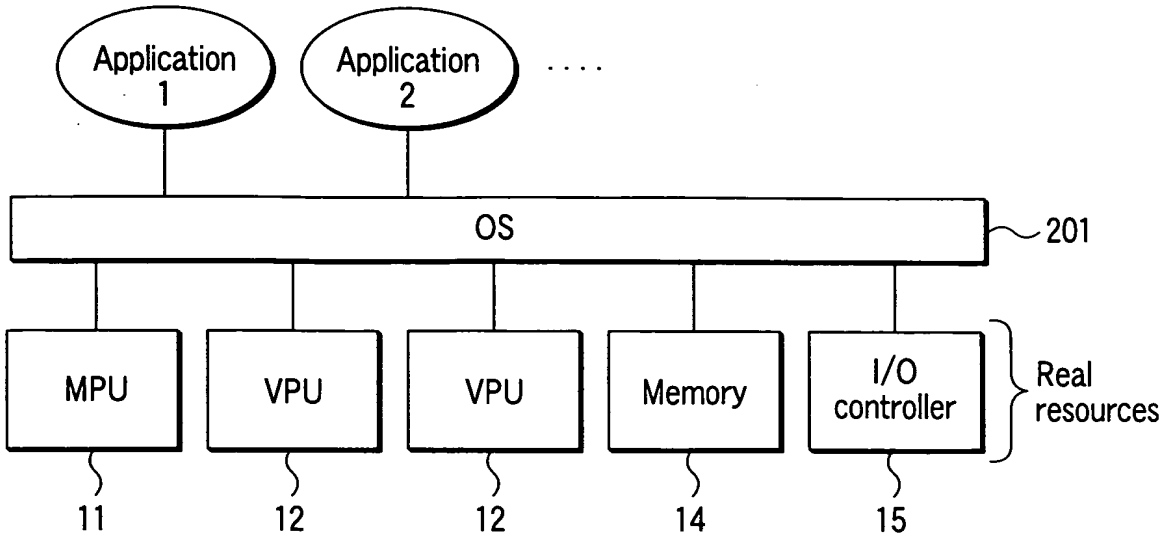
Application 1
Application 2
OS
201
MPU
VPU
VPU
Memory
I/O controller
Real resources
11
12
12
14
15

FIG. 12

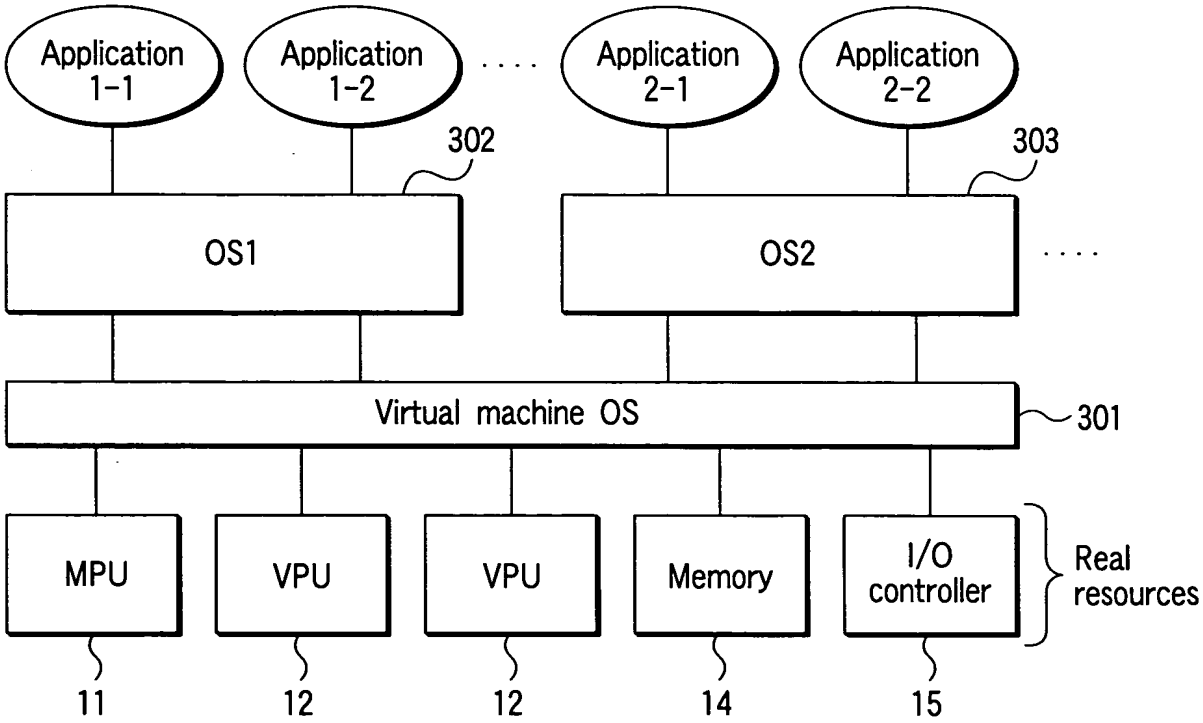
Application 1-1
Application 1-2
Application 2-1
Application 2-2
302
303
OS1
OS2
Virtual machine OS
301
MPU
VPU
VPU
Memory
I/O controller
Real resources
11
12
12
14
15

FIG. 13

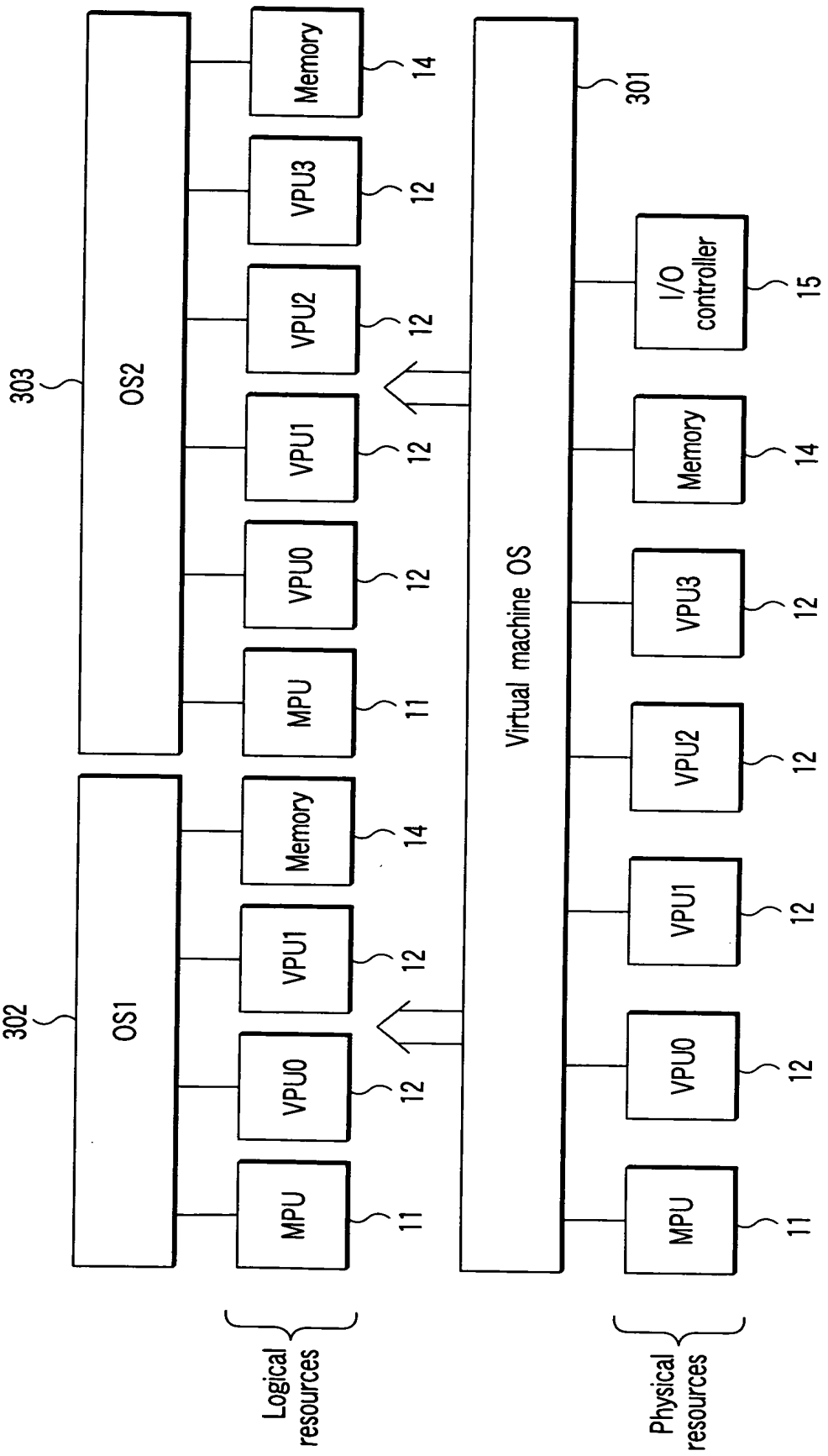
302
OS1
303
OS2
MPU
11
VPU0
12
VPU1
12
Memory
14
VPU2
12
VPU3
12
Memory
14
MPU
11
Logical resources
Virtual machine OS
301
MPU
11
VPU0
12
VPU1
12
VPU2
12
VPU3
12
Memory
14
I/O controller
15
Physical resources

FIG. 14

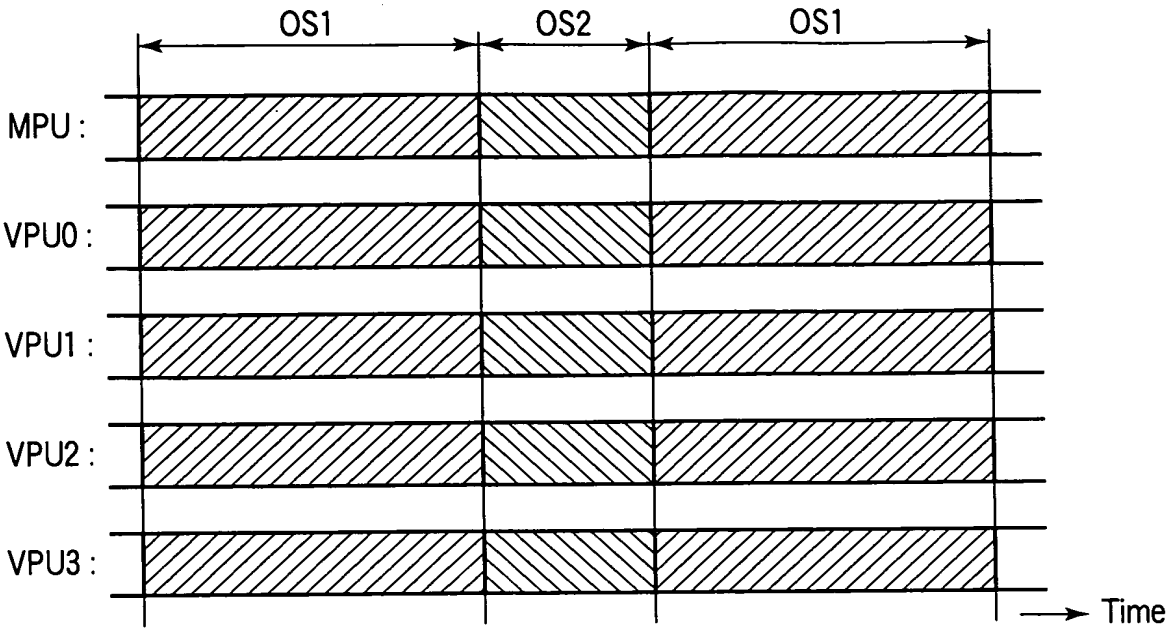
OS1
OS2
OS1
MPU :
VPU0 :
VPU1 :
VPU2 :
VPU3 :
Time

FIG. 15

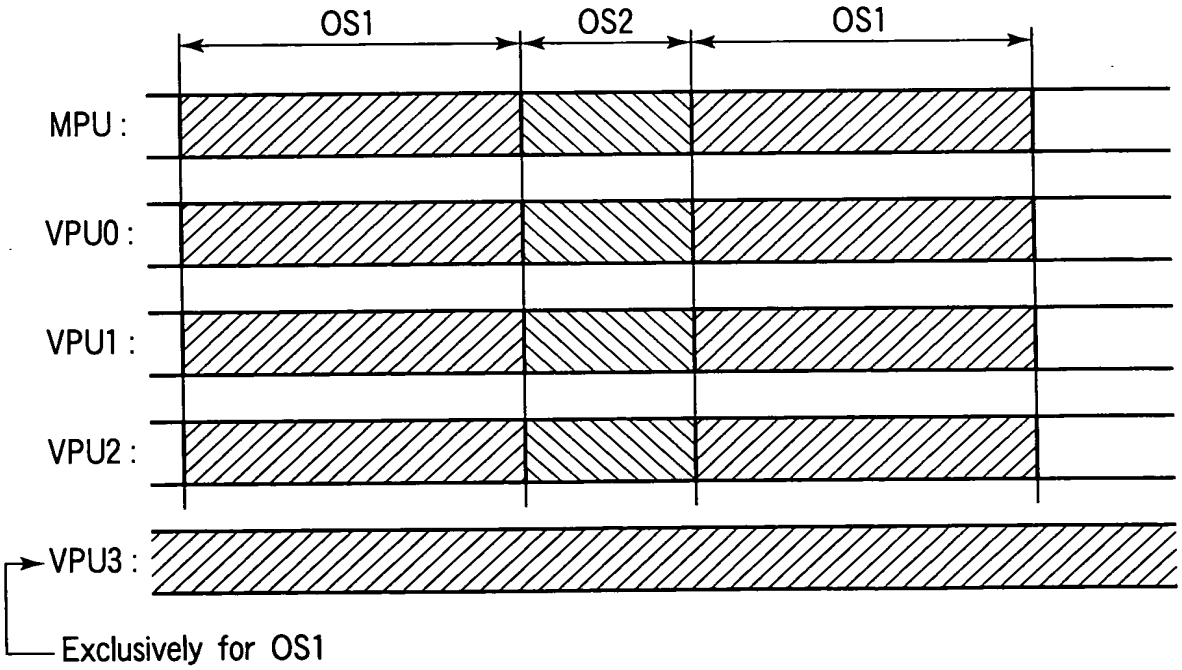
OS1
OS2
OS1
MPU :
VPU0 :
VPU1 :
VPU2 :
VPU3 :
Exclusively for OS1

FIG. 16

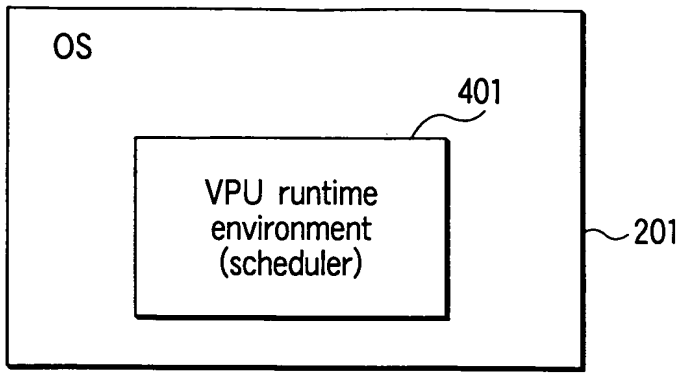


FIG. 17

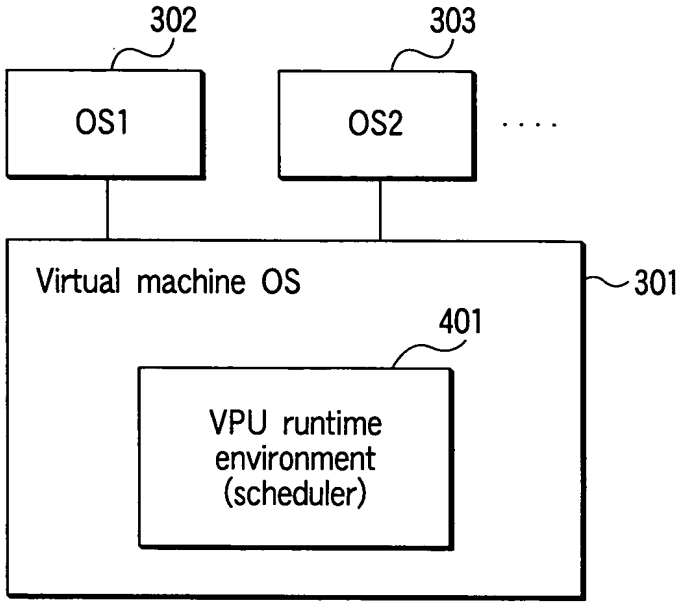


FIG. 18

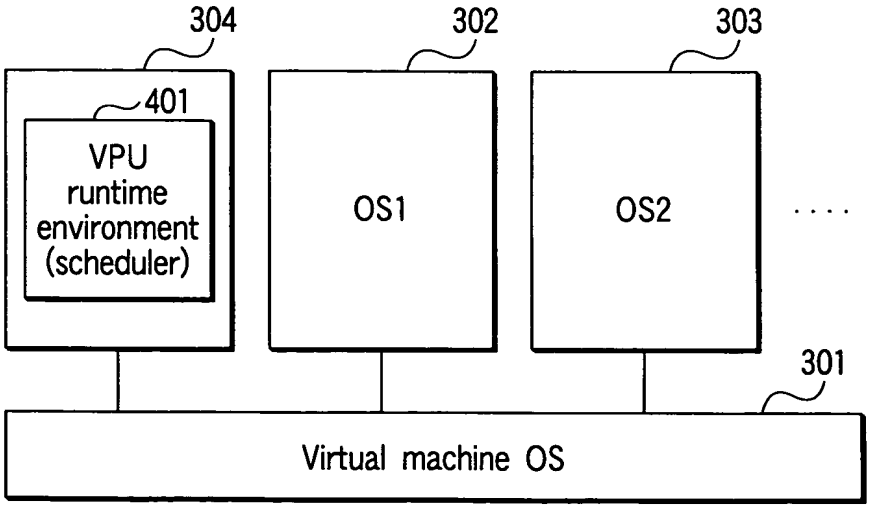


FIG. 19

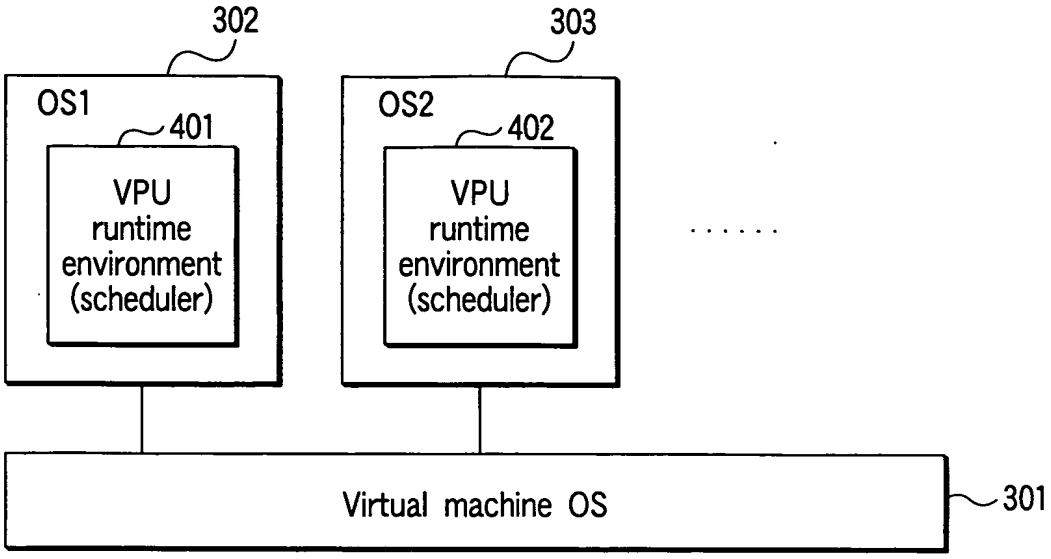
302
303
OS1
401
VPU runtime environment (scheduler)
OS2
402
VPU runtime environment (scheduler)
Virtual machine OS
301

FIG. 20

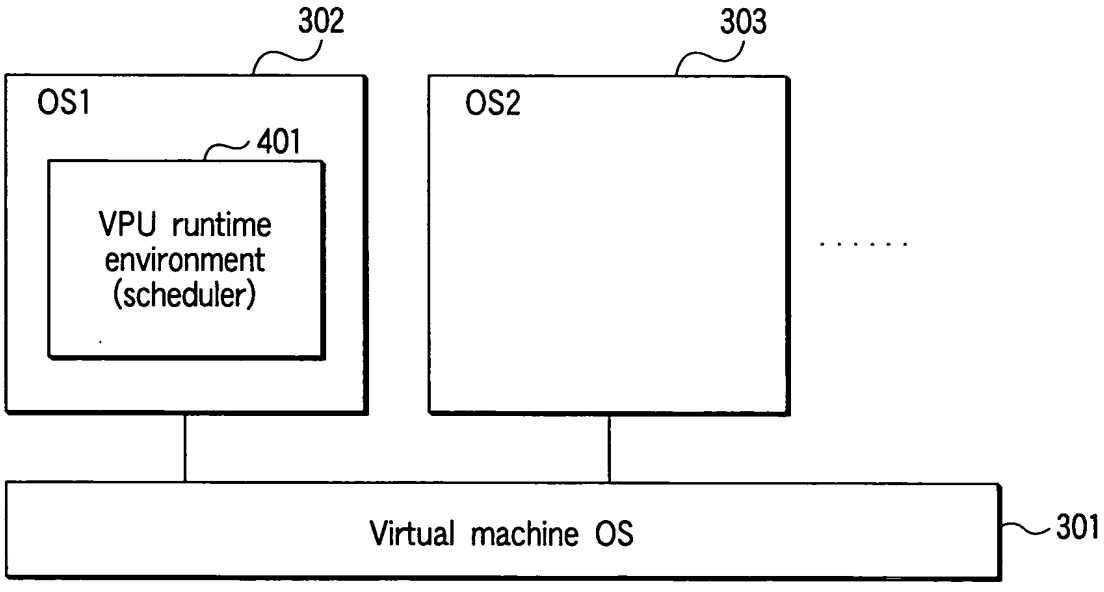
302
303
OS1
401
VPU runtime environment (scheduler)
OS2
Virtual machine OS
301

FIG. 21

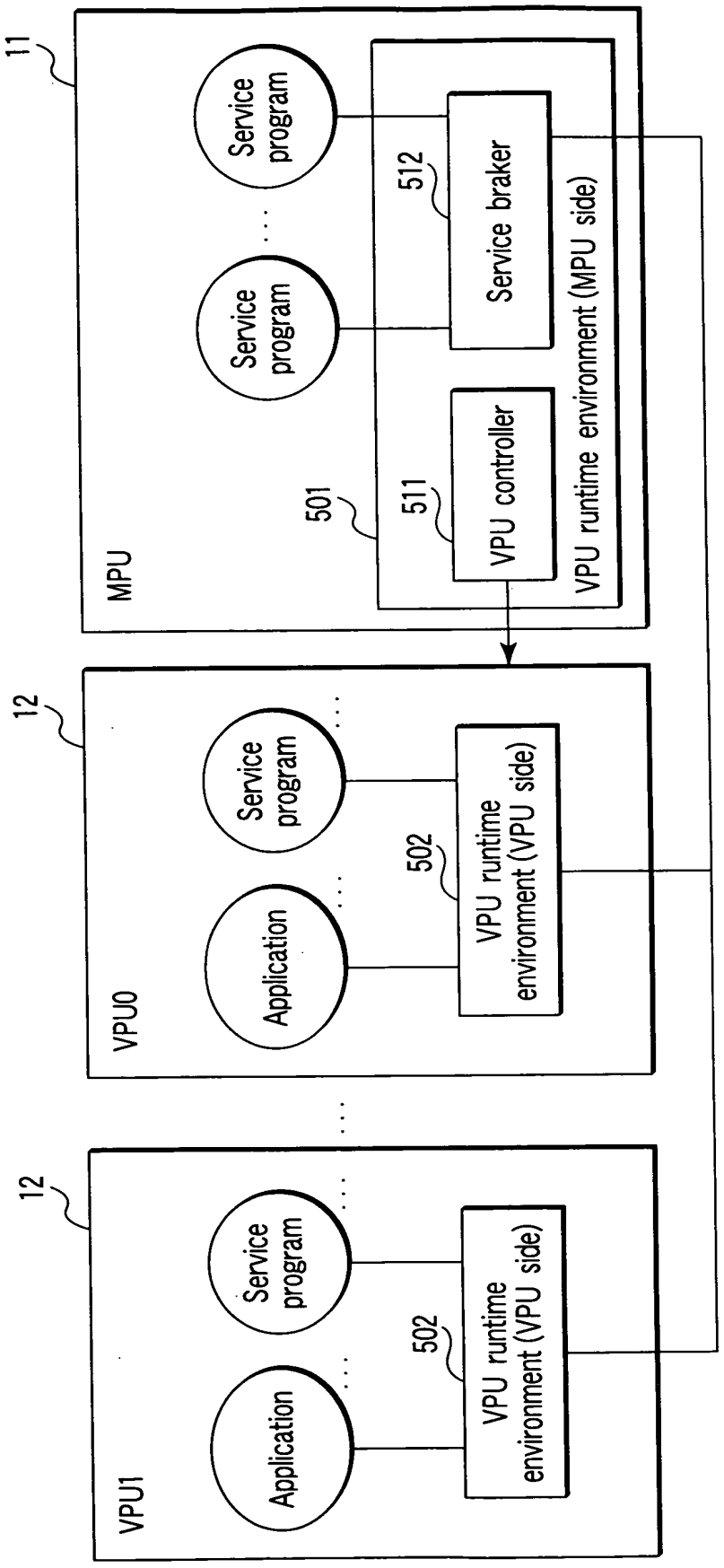
11
MPU
Service program
Service program
512
Service braker
501
511
VPU controller
VPU runtime environment (MPU side)
12
VPU0
Service program
Application
502
VPU runtime environment (VPU side)
12
VPU1
Service program
Application
502
VPU runtime environment (VPU side)

FIG. 22

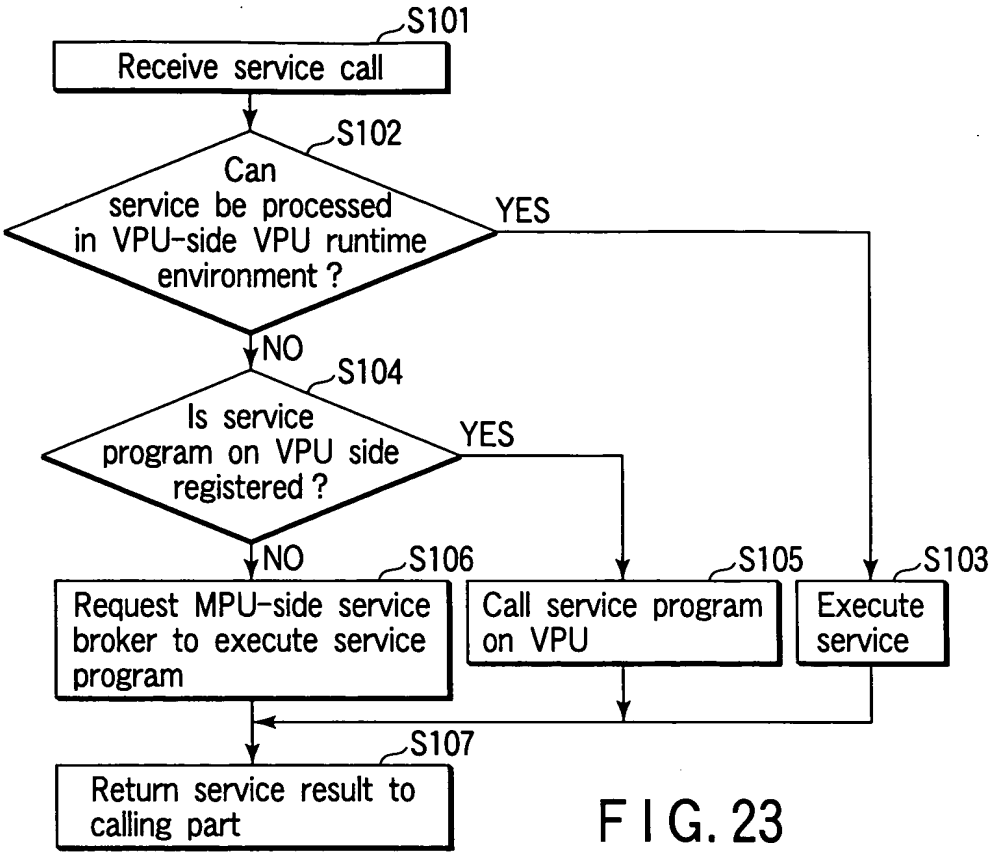
S101
Receive service call
S102
Can service be processed in VPU-side VPU runtime environment ?
YES
NO
S104
Is service program on VPU side registered ?
YES
NO
S106
Request MPU-side service broker to execute service program
S105
Call service program on VPU
S103
Execute service
S107
Return service result to calling part

FIG. 23

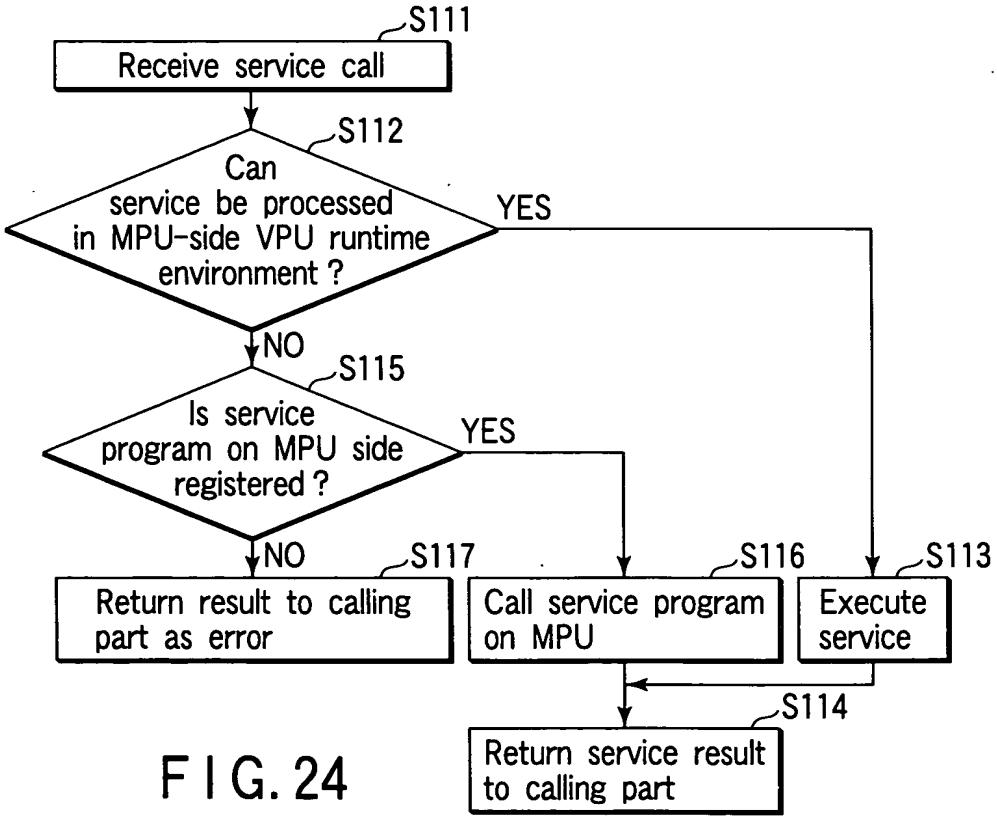
S111
Receive service call
S112
Can service be processed in MPU-side VPU runtime environment ?
YES
NO
S115
Is service program on MPU side registered ?
YES
NO
S117
Return result to calling part as error
S116
Call service program on MPU
S113
Execute service
S114
Return service result to calling part

FIG. 24

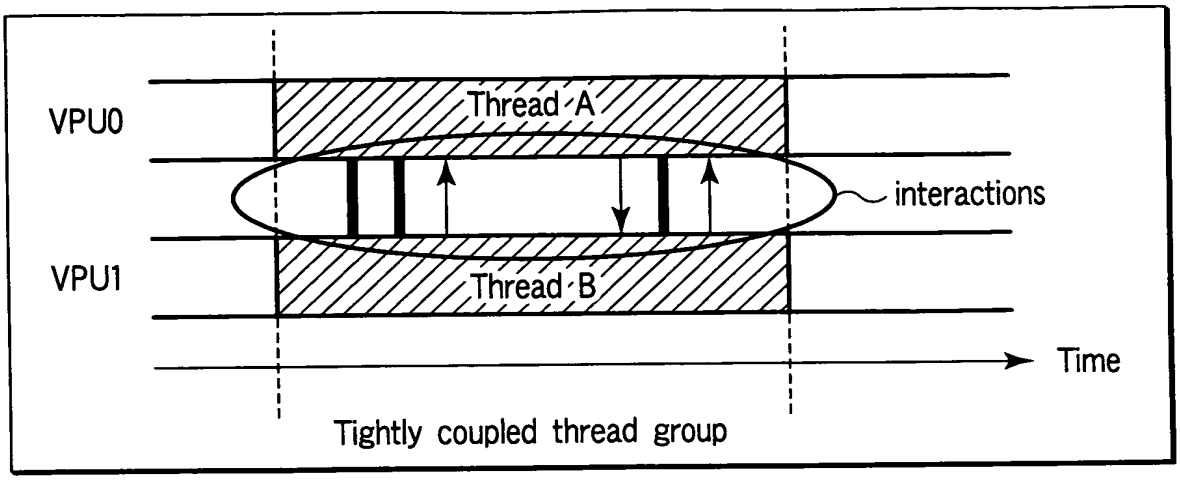
VPU0
Thread A
interactions
VPU1
Thread B
Time
Tightly coupled thread group

FIG. 25

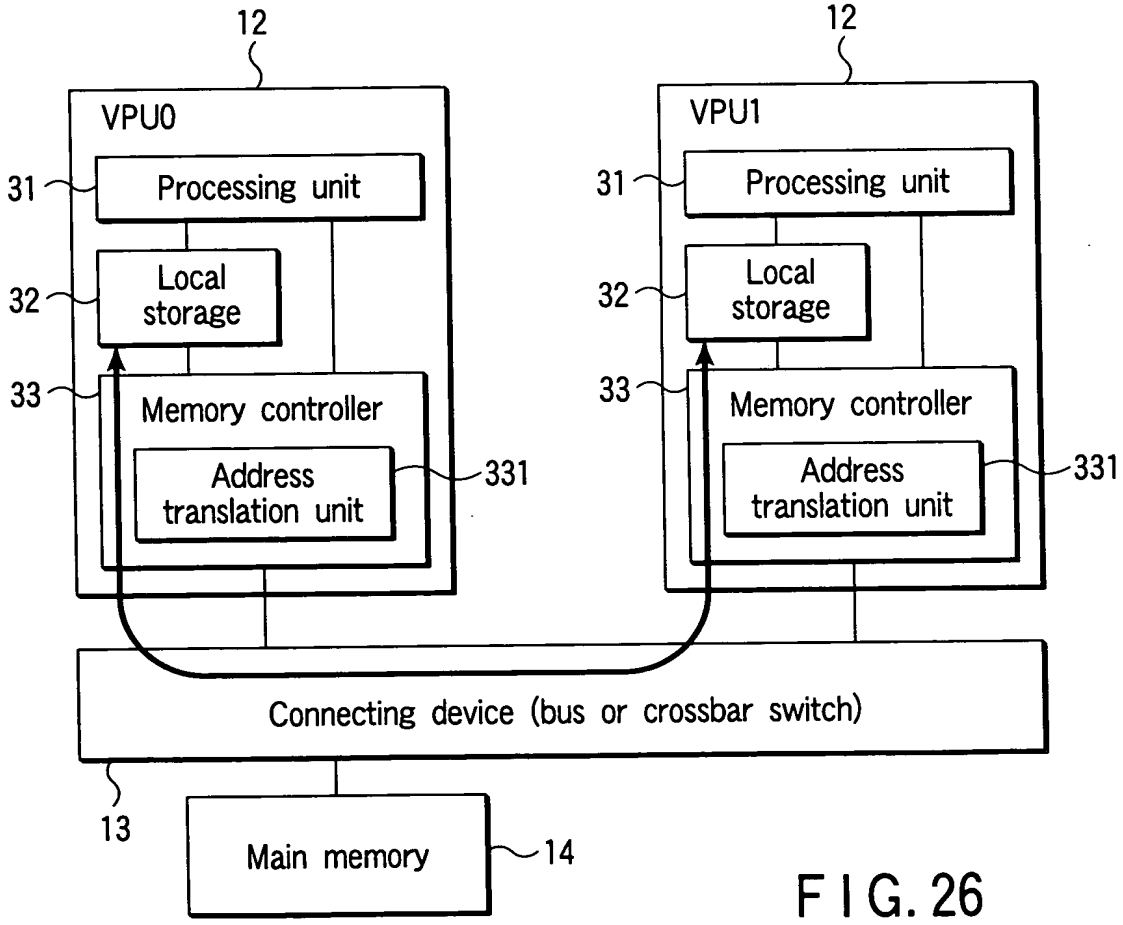
12
12
VPU0
VPU1
31
Processing unit
31
Processing unit
32
Local storage
32
Local storage
33
Memory controller
33
Memory controller
Address translation unit
331
Address translation unit
331
Connecting device (bus or crossbar switch)
13
Main memory
14

FIG. 26

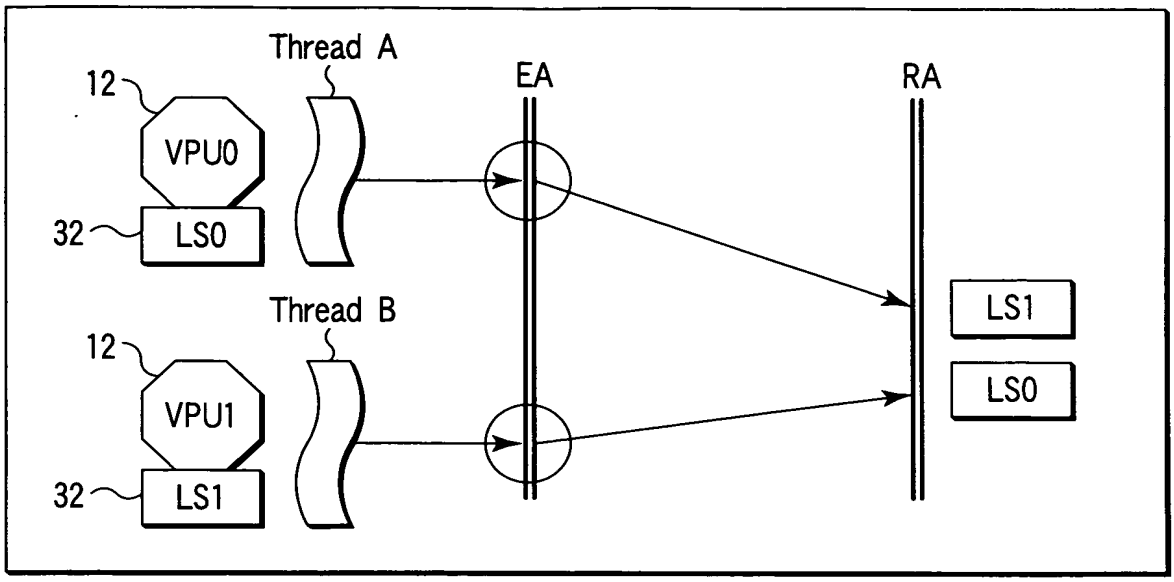
Thread A
EA
RA
12
VPU0
32
LS0
Thread B
12
VPU1
32
LS1
LS1
LS0

FIG. 27

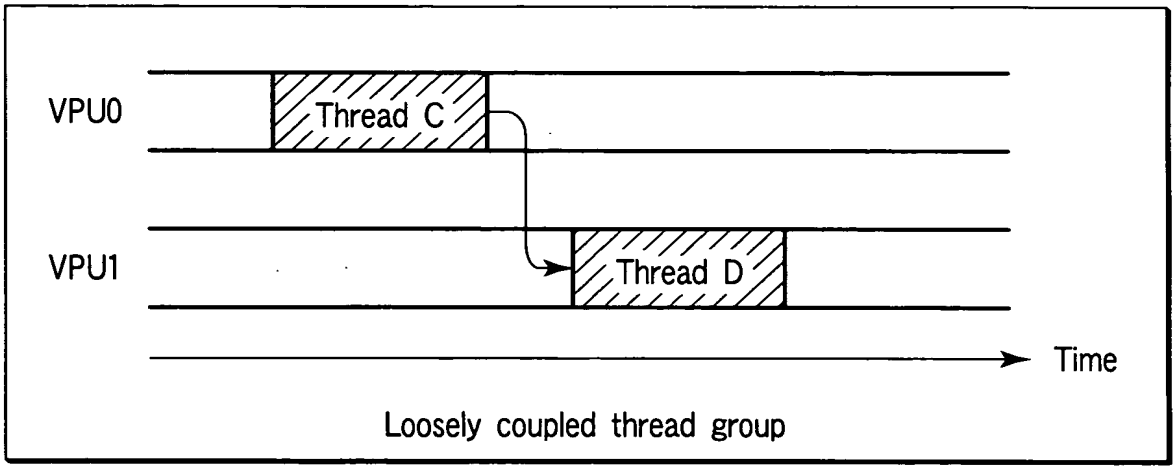
VPU0
Thread C
VPU1
Thread D
Time
Loosely coupled thread group

FIG. 28

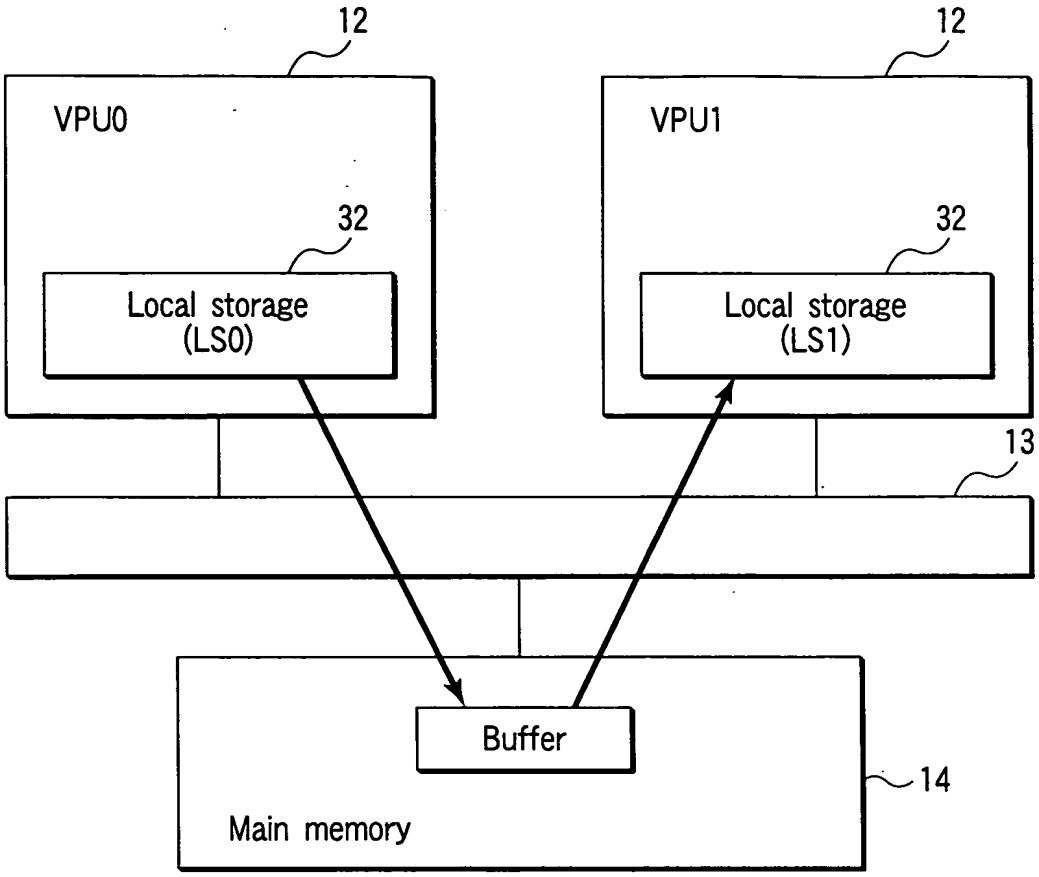
12
12
VPU0
VPU1
32
32
Local storage
(LS0)
Local storage
(LS1)
13
Buffer
14
Main memory

FIG. 29

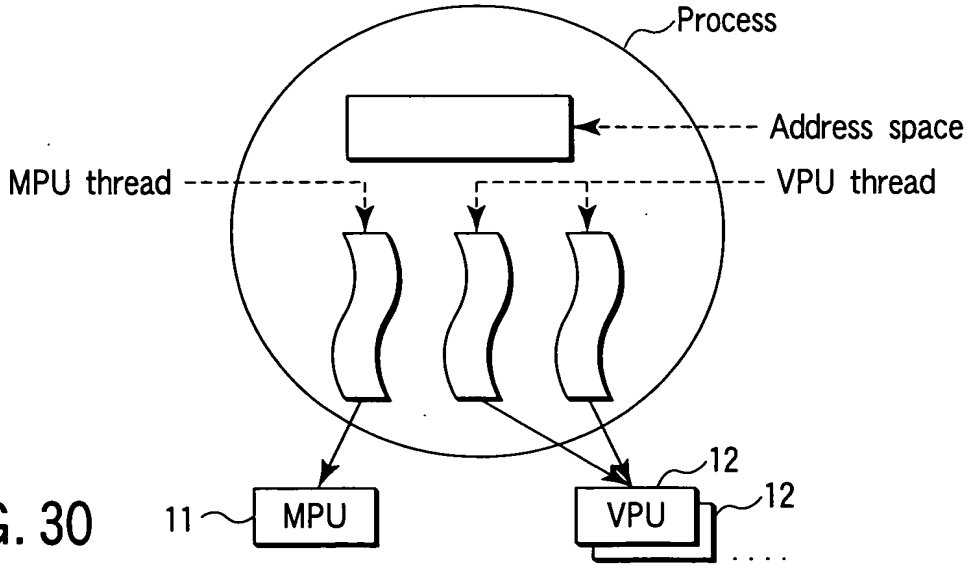
Process
Address space
MPU thread
VPU thread
12
11
MPU
VPU
12

FIG. 30

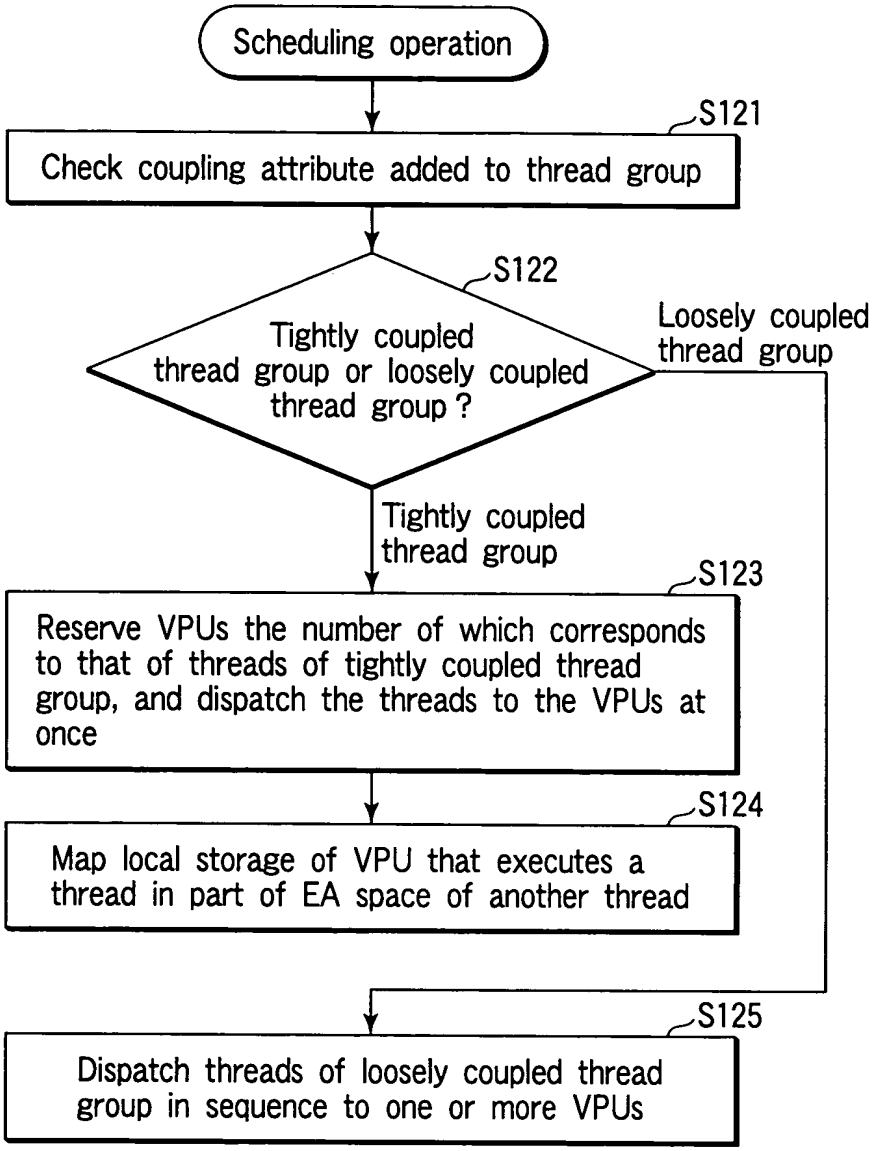
Scheduling operation
S121
Check coupling attribute added to thread group
S122
Tightly coupled thread group or loosely coupled thread group ?
Loosely coupled thread group
Tightly coupled thread group
S123
Reserve VPUs the number of which corresponds to that of threads of tightly coupled thread group, and dispatch the threads to the VPUs at once
S124
Map local storage of VPU that executes a thread in part of EA space of another thread
S125
Dispatch threads of loosely coupled thread group in sequence to one or more VPUs

FIG. 31

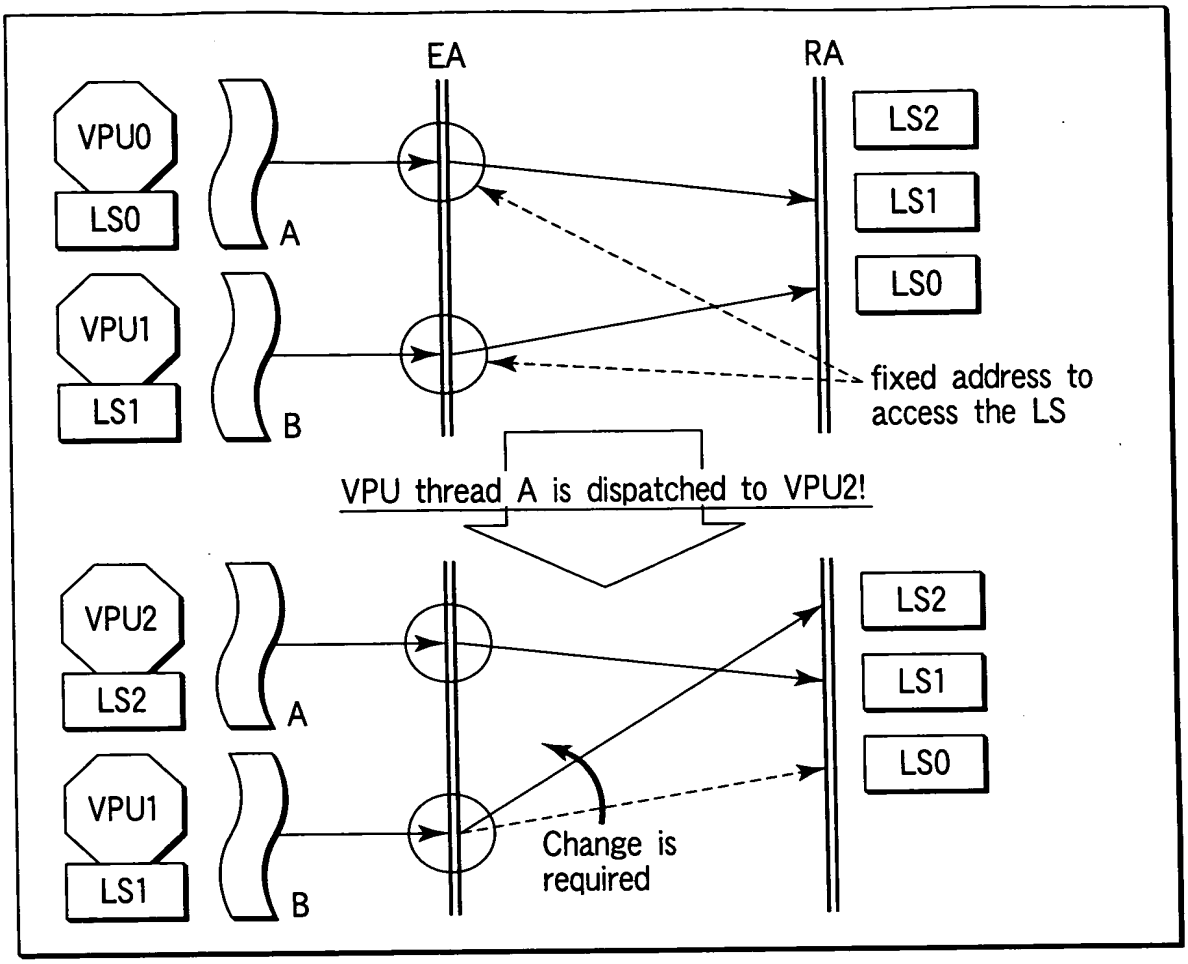
EA
RA
VPU0
LS0
A
VPU1
LS1
B
LS2
LS1
LS0
fixed address to access the LS
VPU thread A is dispatched to VPU2!
VPU2
LS2
A
VPU1
LS1
B
LS2
LS1
LS0
Change is required

FIG. 32

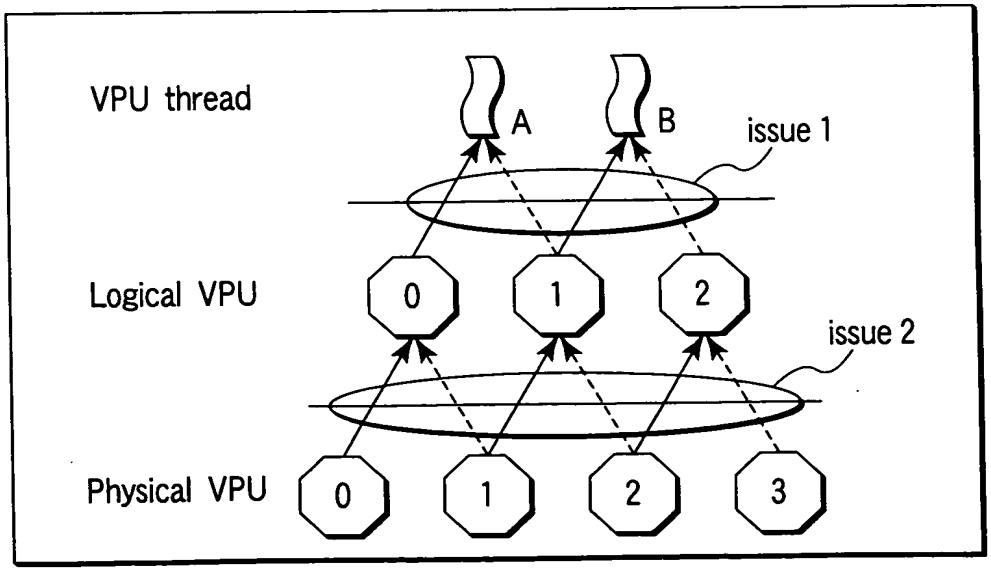
VPU thread
A
B
issue 1
Logical VPU
0
1
2
issue 2
Physical VPU
0
1
2
3

FIG. 33

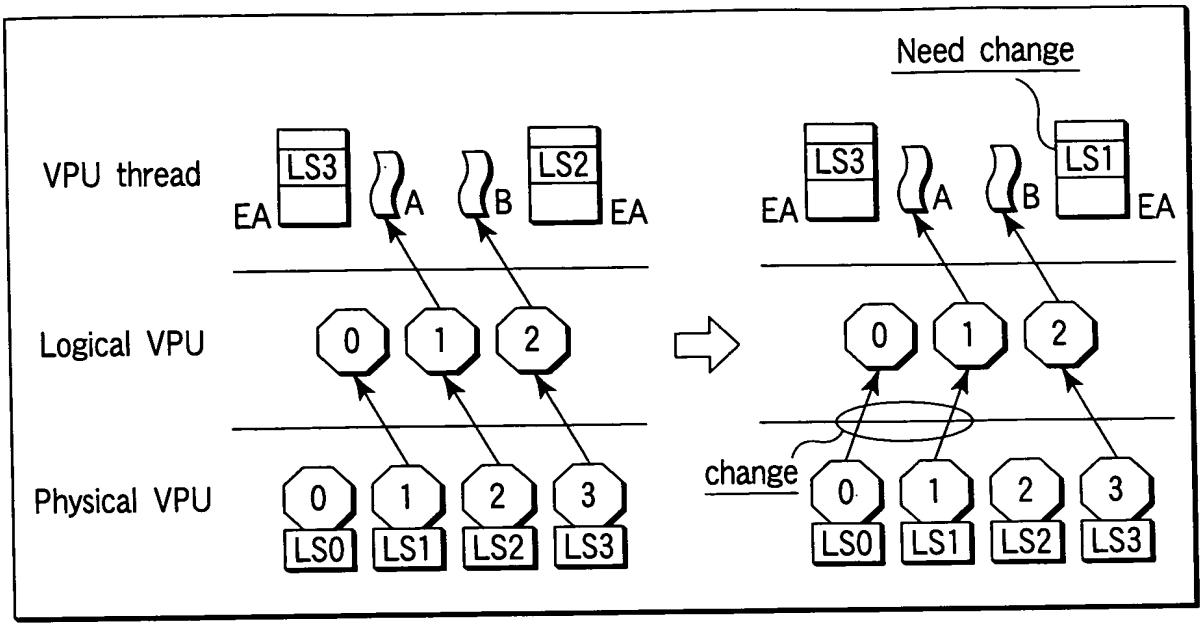
Need change
VPU thread
EA
LS3
A
B
LS2
EA
LS3
A
B
LS1
EA
Logical VPU
0
1
2
0
1
2
Physical VPU
0
1
2
3
LS0
LS1
LS2
LS3
change
0
1
2
3
LS0
LS1
LS2
LS3

FIG. 34

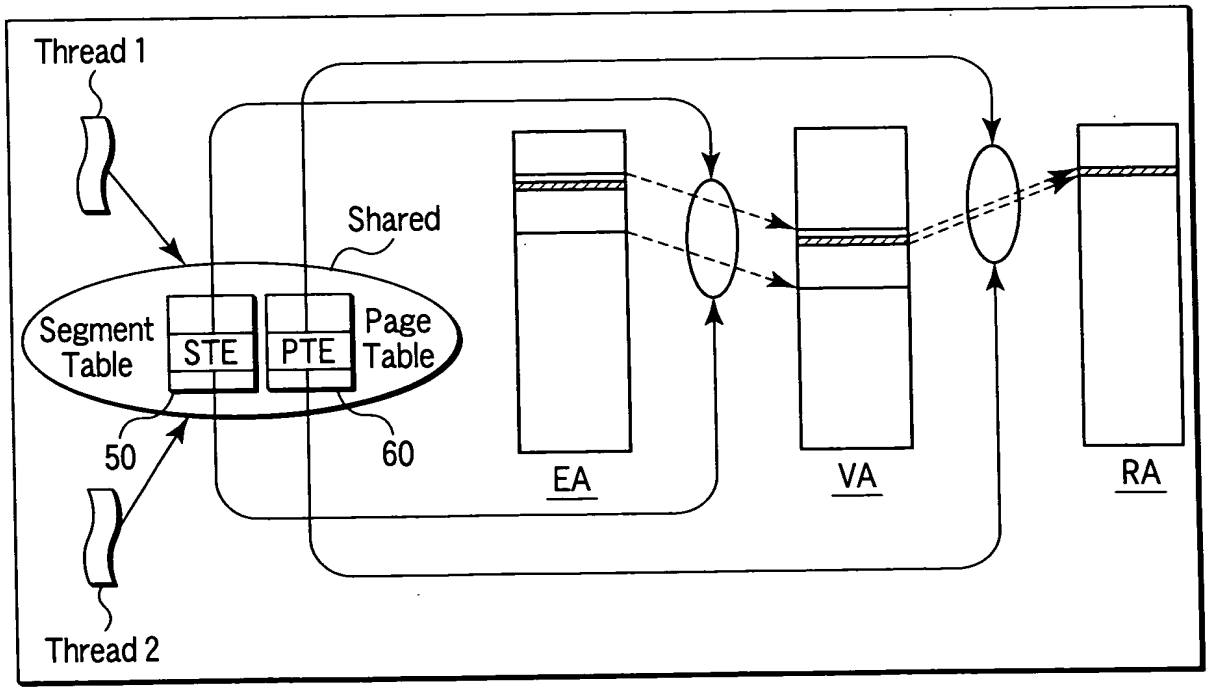
Thread 1
Shared
Segment Table
STE
PTE
Page Table
50
60
EA
VA
RA
Thread 2

FIG. 35

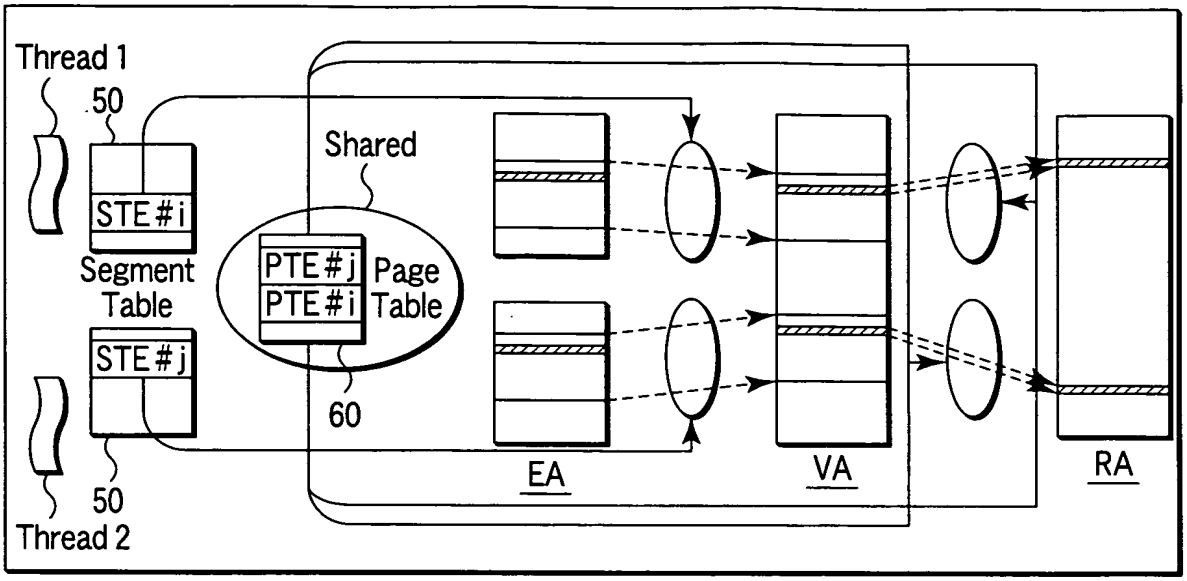
Thread 1
50
STE#i
Segment Table
STE#j
50
Thread 2
Shared
PTE#j
PTE#i
Page Table
60
EA
VA
RA

FIG. 36

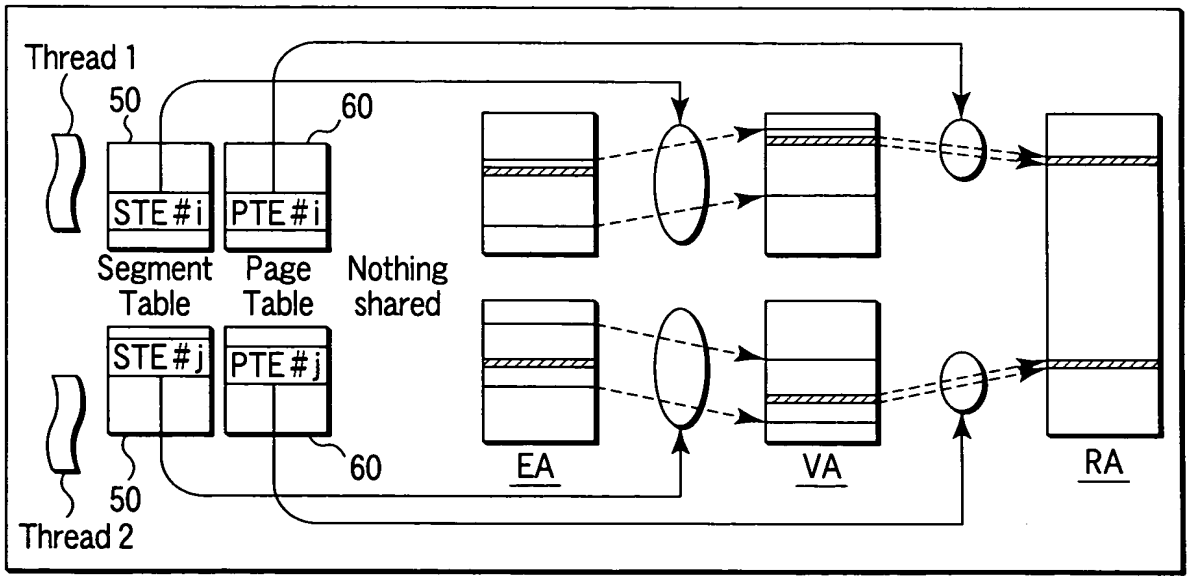
Thread 1
50
60
STE#i
PTE#i
Segment Table
Page Table
Nothing shared
STE#j
PTE#j
50
60
Thread 2
EA
VA
RA

FIG. 37

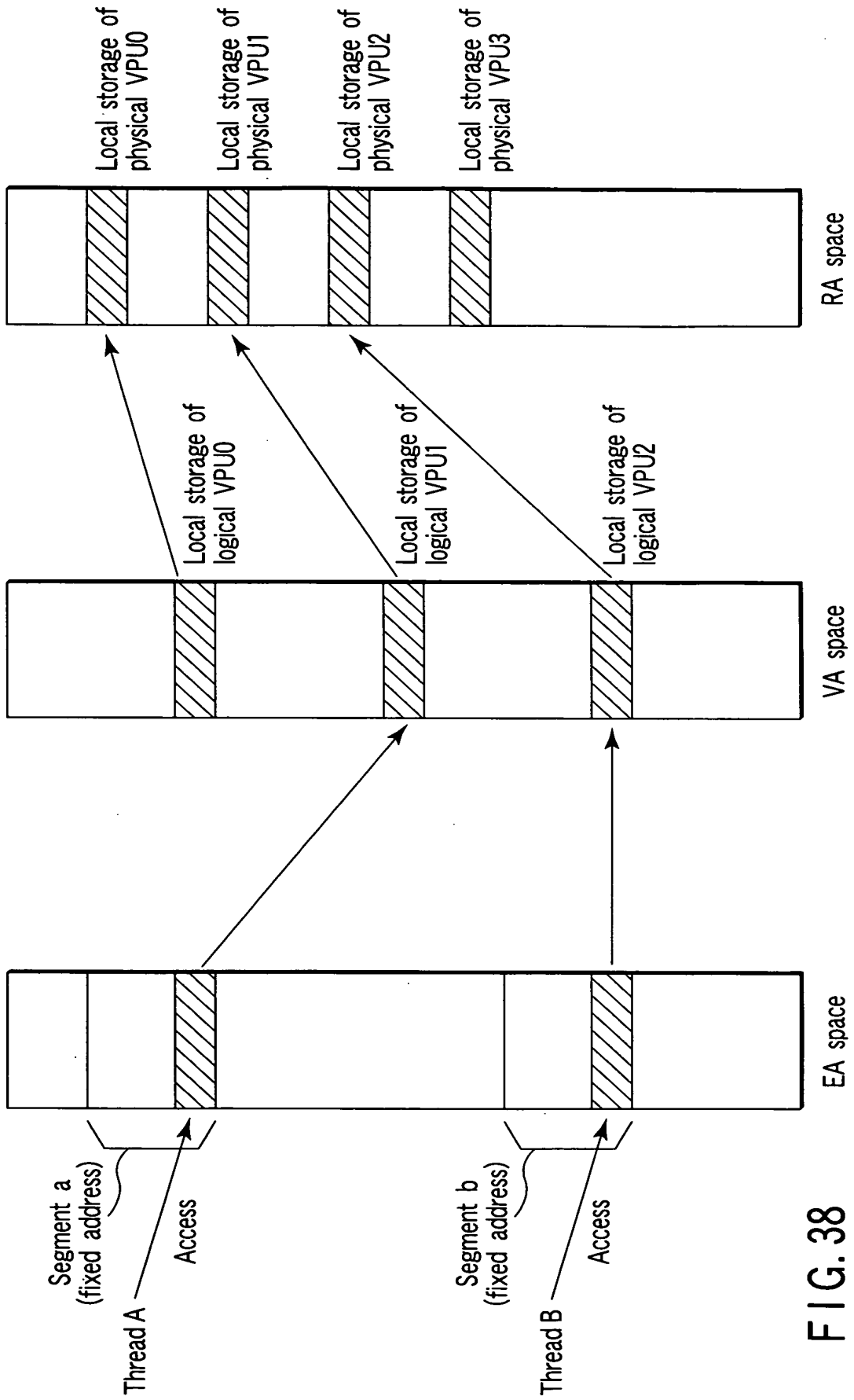
Local storage of physical VPU0
Local storage of physical VPU1
Local storage of physical VPU2
Local storage of physical VPU3
RA space
Local storage of logical VPU0
Local storage of logical VPU1
Local storage of logical VPU2
VA space
EA space
Segment a (fixed address)
Access
Thread A
Segment b (fixed address)
Access
Thread B

FIG. 38

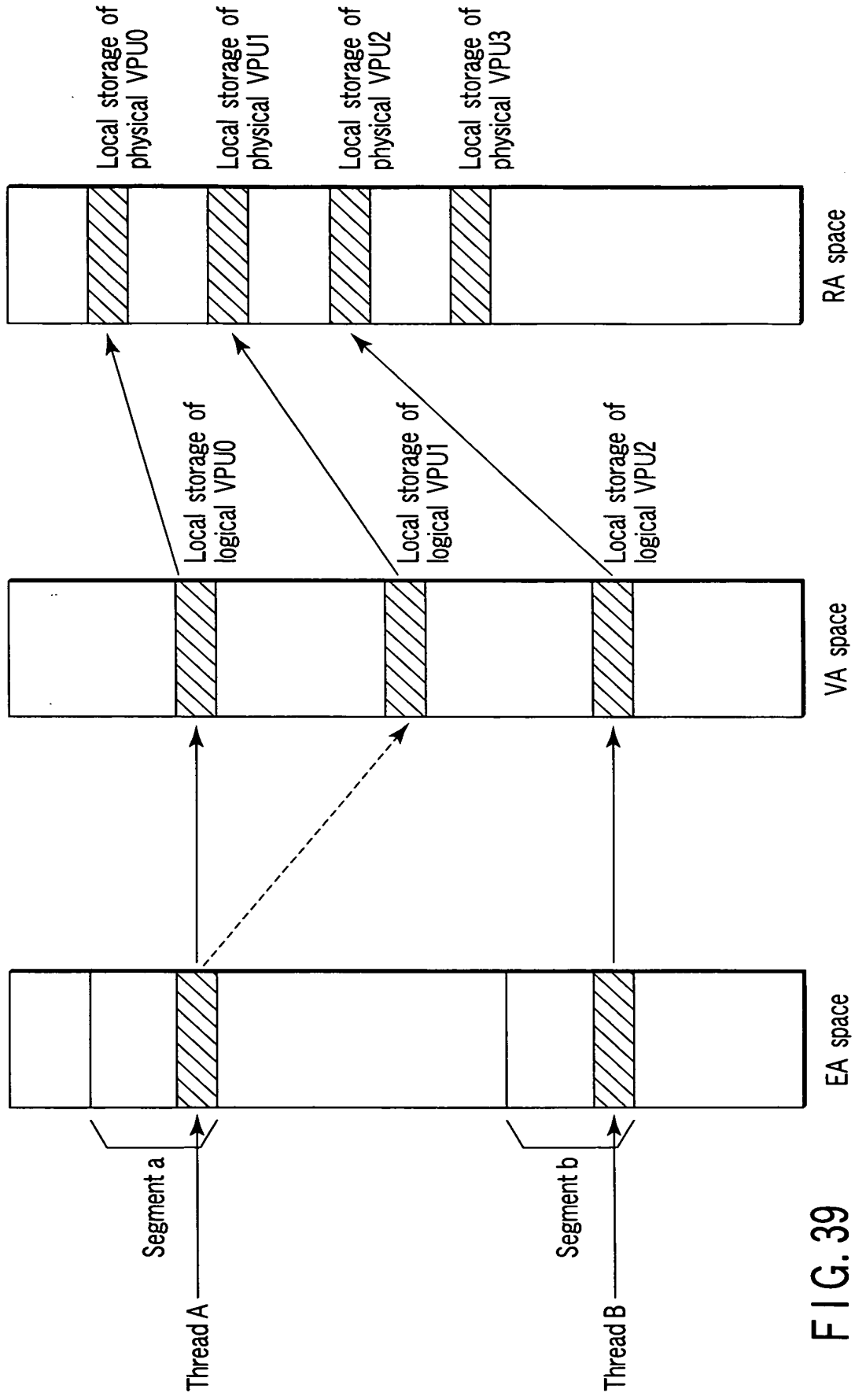
Local storage of physical VPU0
Local storage of physical VPU1
Local storage of physical VPU2
Local storage of physical VPU3
RA space
Local storage of logical VPU0
Local storage of logical VPU1
Local storage of logical VPU2
VA space
EA space
Segment a
Segment b
Thread A
Thread B

FIG. 39

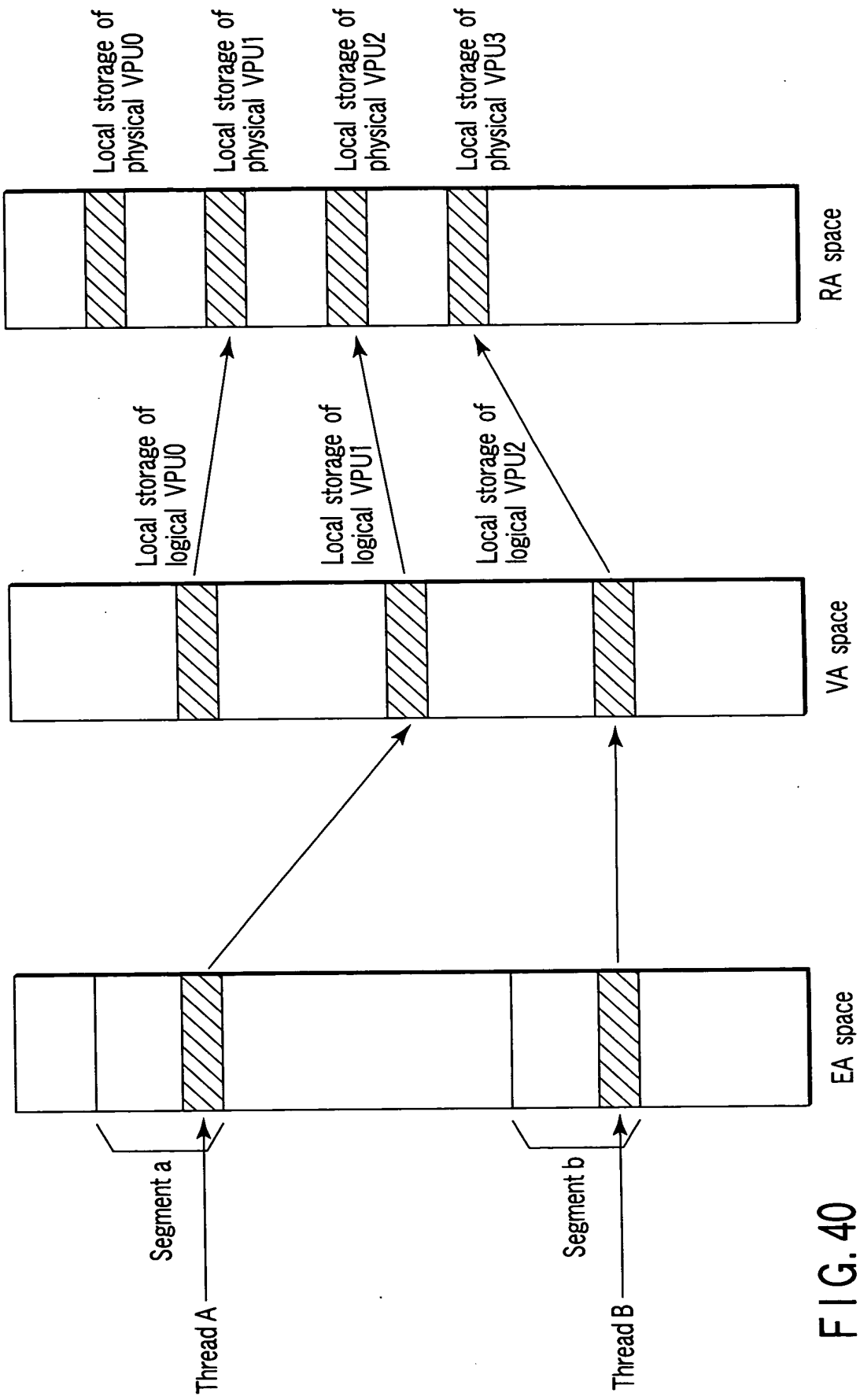
Local storage of physical VPU0
Local storage of physical VPU1
Local storage of physical VPU2
Local storage of physical VPU3
RA space
Local storage of logical VPU0
Local storage of logical VPU1
Local storage of logical VPU2
VA space
EA space
Segment a
Segment b
Thread A
Thread B

FIG. 40

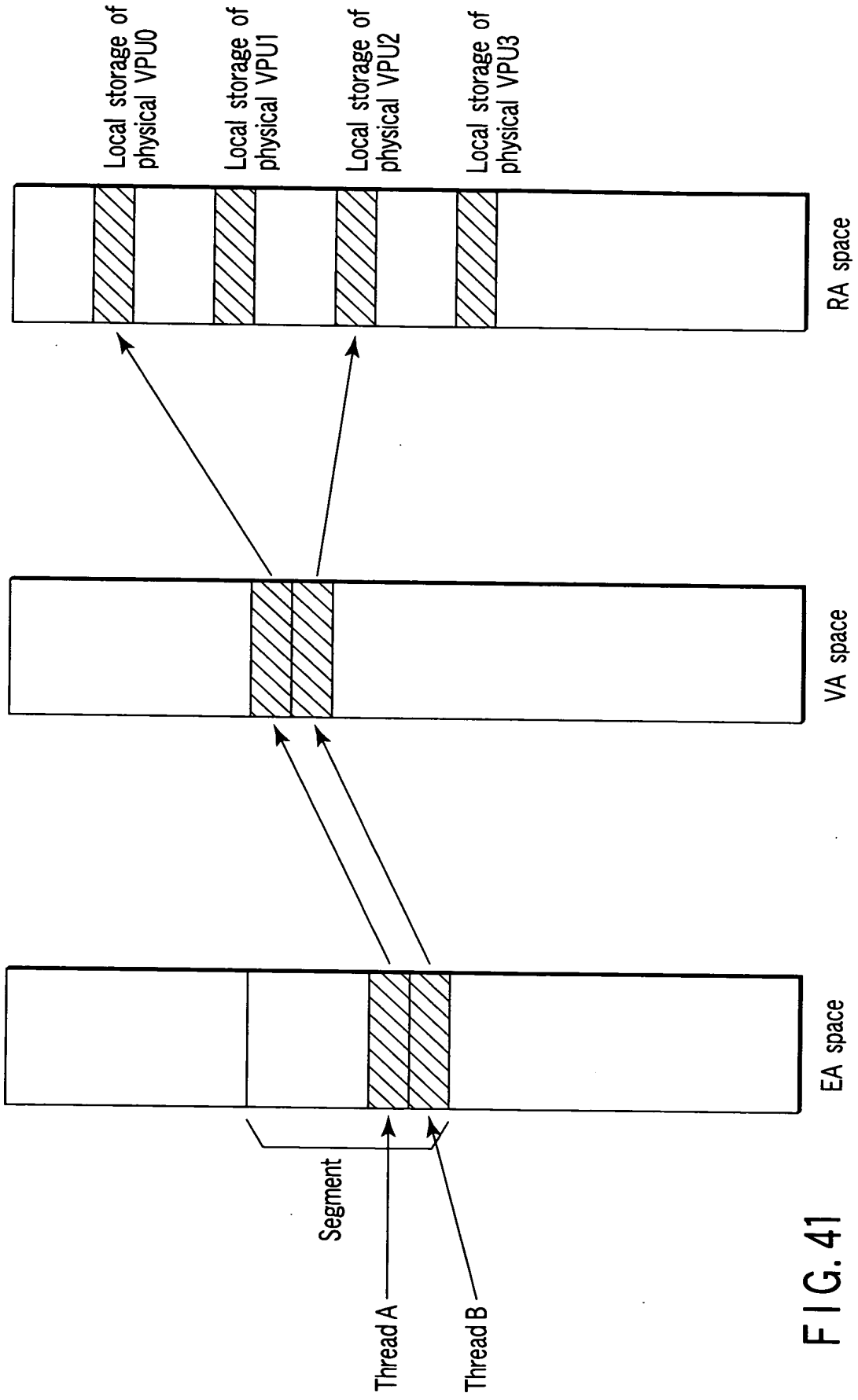
Local storage of physical VPU0
Local storage of physical VPU1
Local storage of physical VPU2
Local storage of physical VPU3
RA space
VA space
EA space
Segment
Thread A
Thread B

FIG. 41

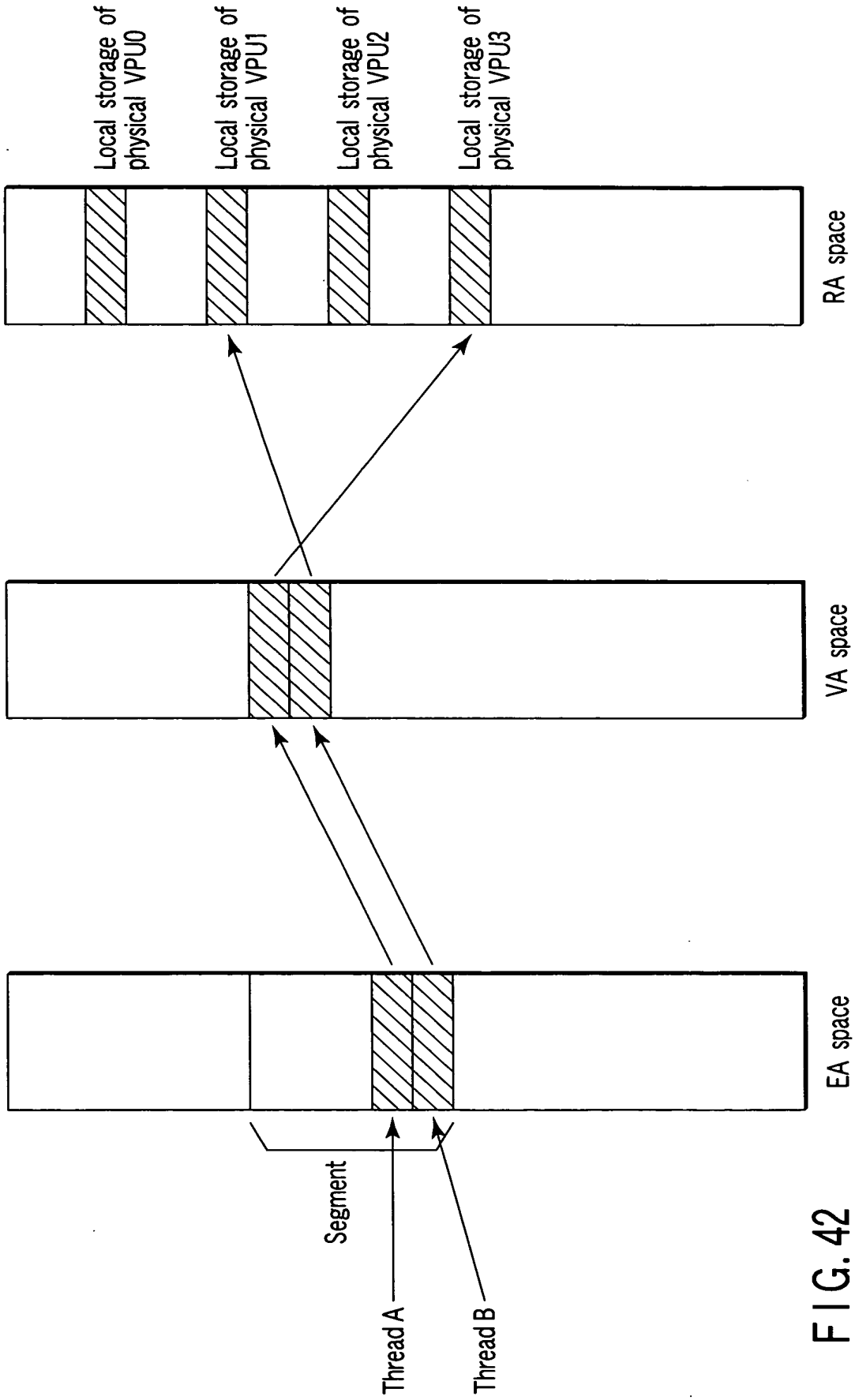
Local storage of physical VPU0
Local storage of physical VPU1
Local storage of physical VPU2
Local storage of physical VPU3
RA space
VA space
EA space
Segment
Thread A
Thread B

FIG. 42

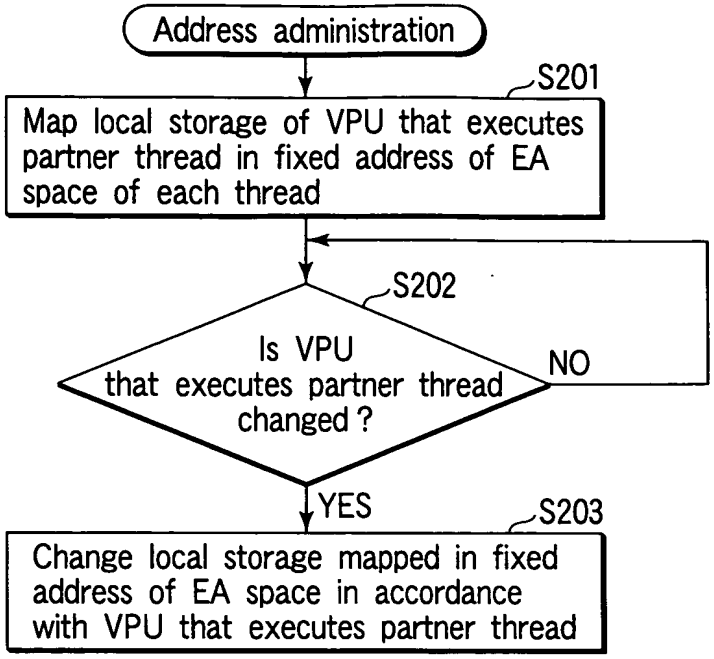
Address administration
S201
Map local storage of VPU that executes partner thread in fixed address of EA space of each thread
S202
Is VPU that executes partner thread changed ?
NO
YES
S203
Change local storage mapped in fixed address of EA space in accordance with VPU that executes partner thread

FIG. 43

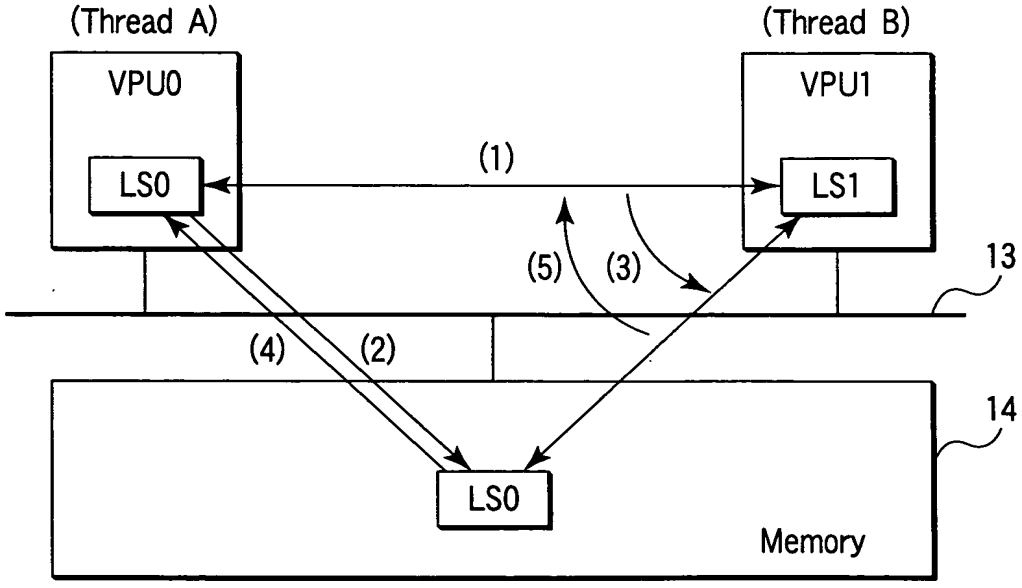
(Thread A)
(Thread B)
VPU0
VPU1
LS0
LS1
(1)
(2)
(3)
(4)
(5)
13
14
LS0
Memory

FIG. 44

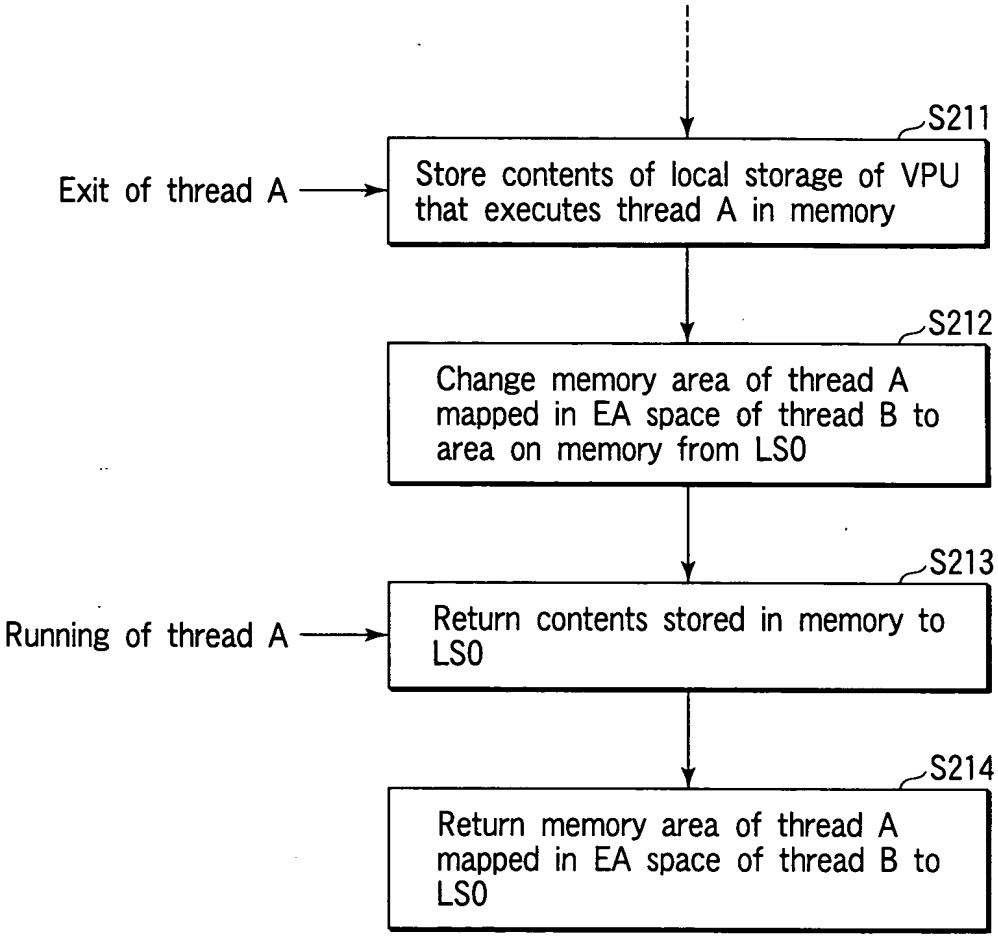
S211
Exit of thread A
Store contents of local storage of VPU that executes thread A in memory
S212
Change memory area of thread A mapped in EA space of thread B to area on memory from LS0
S213
Running of thread A
Return contents stored in memory to LS0
S214
Return memory area of thread A mapped in EA space of thread B to LS0

FIG. 45

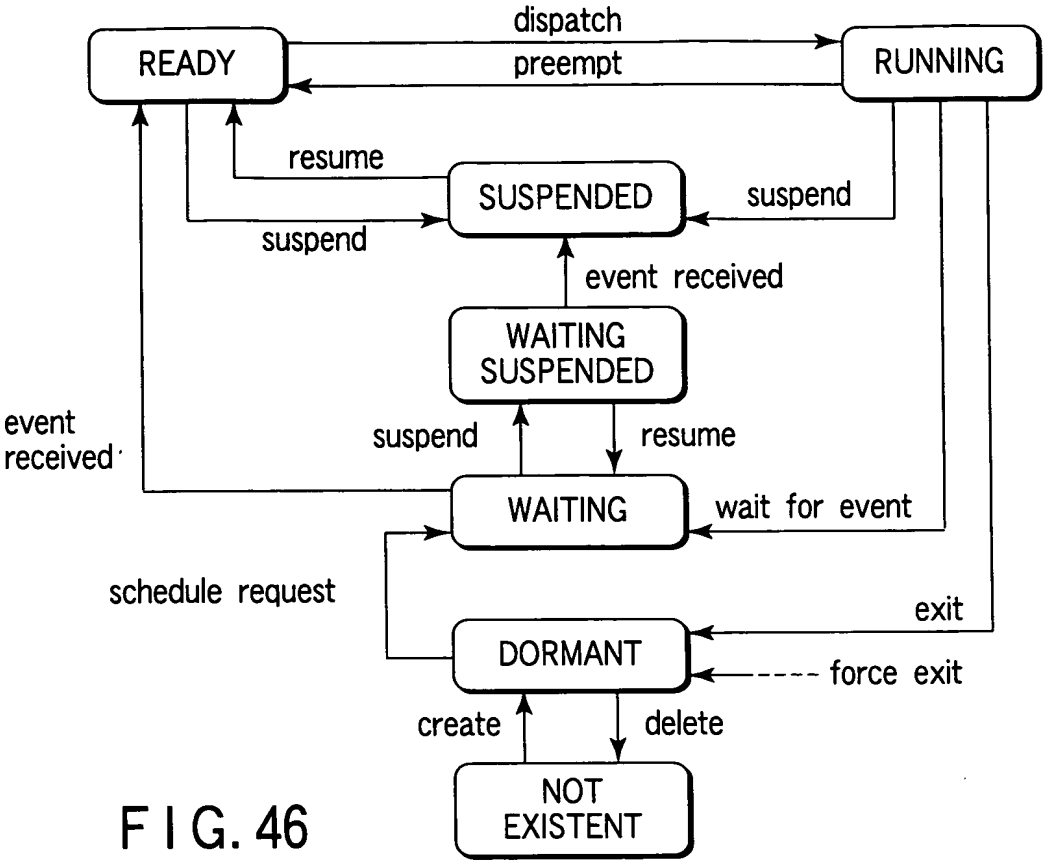


FIG. 46

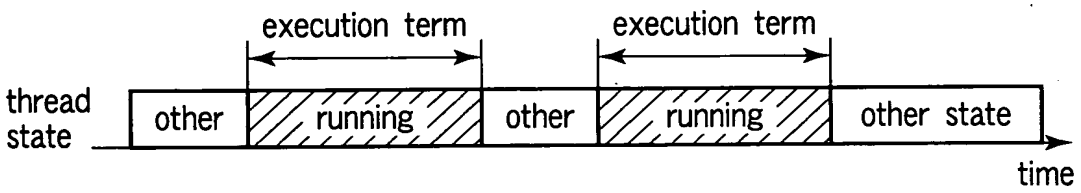


FIG. 47

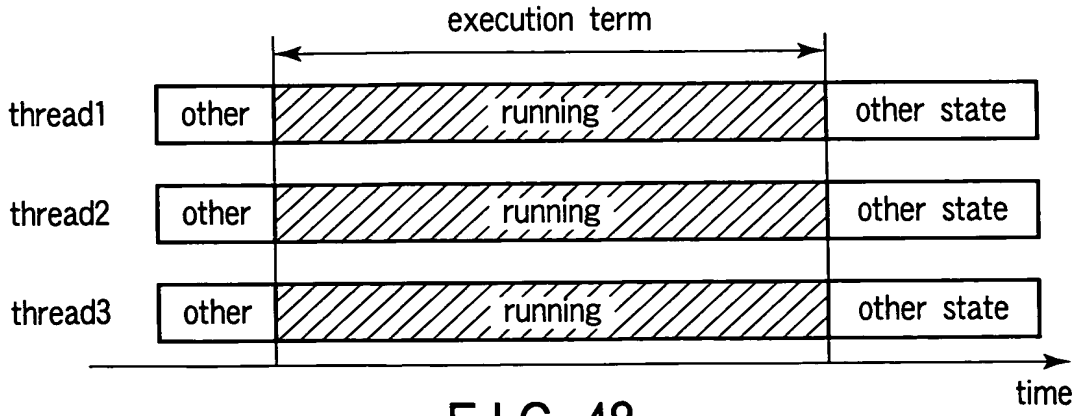


FIG. 48

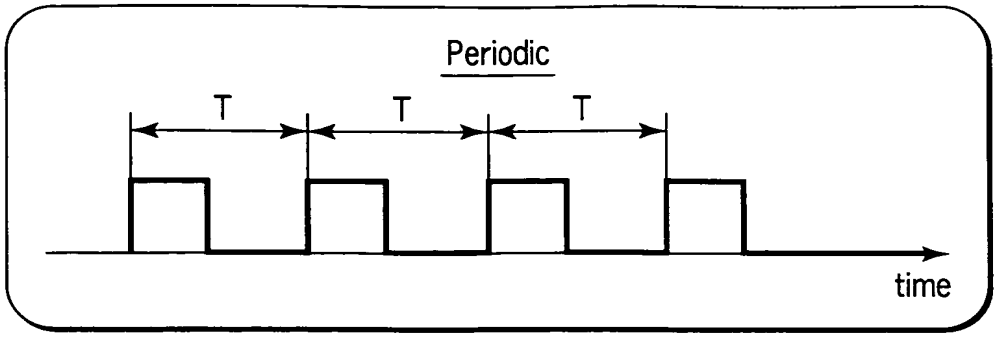


FIG. 49

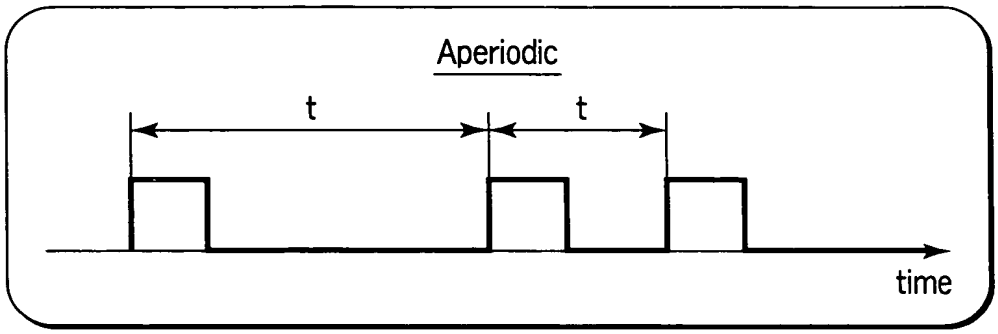


FIG. 50

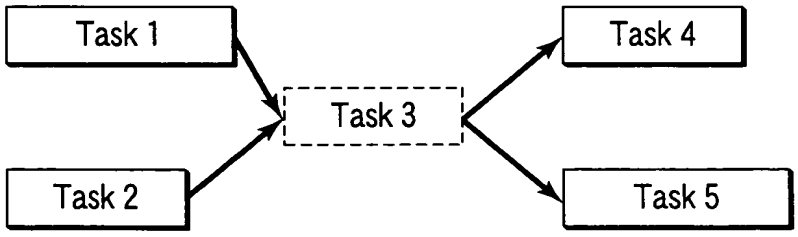


FIG. 51

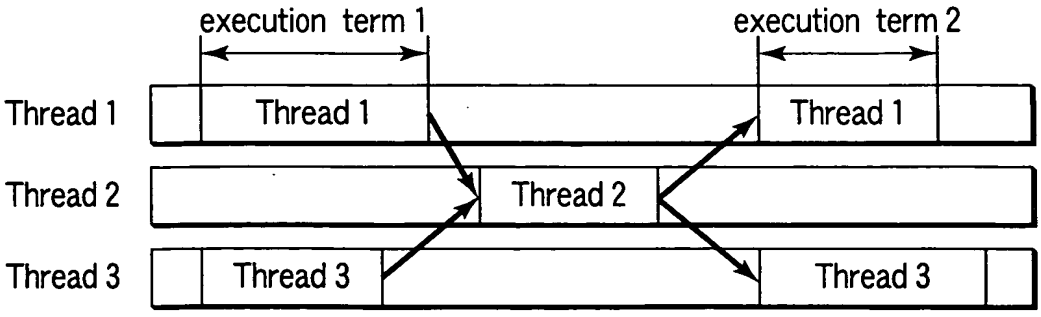


FIG. 52

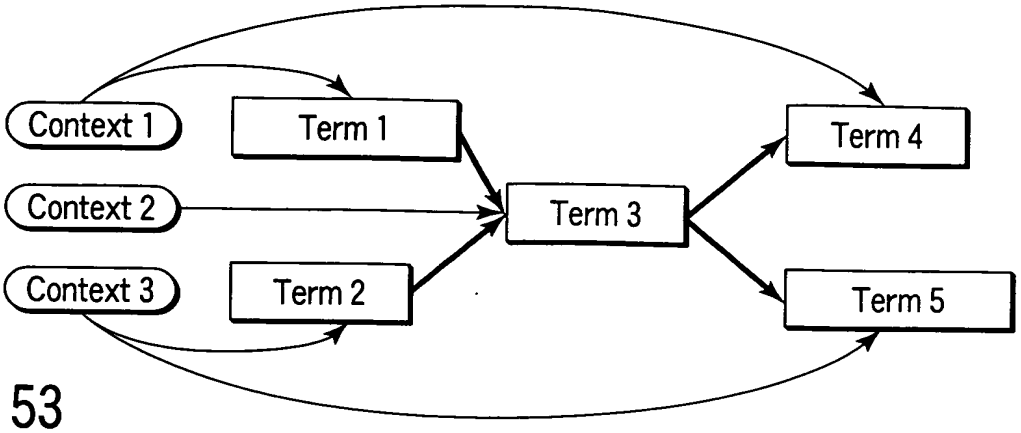


FIG. 53

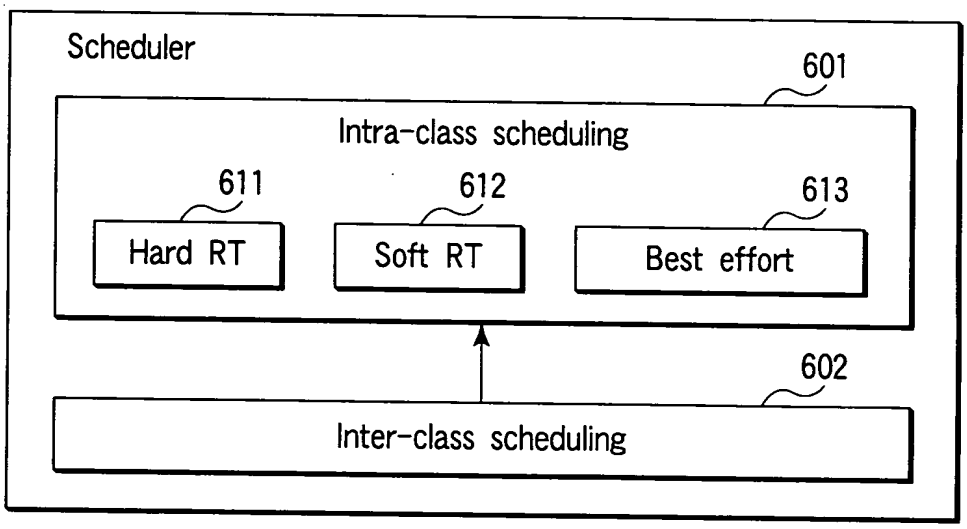


FIG. 54

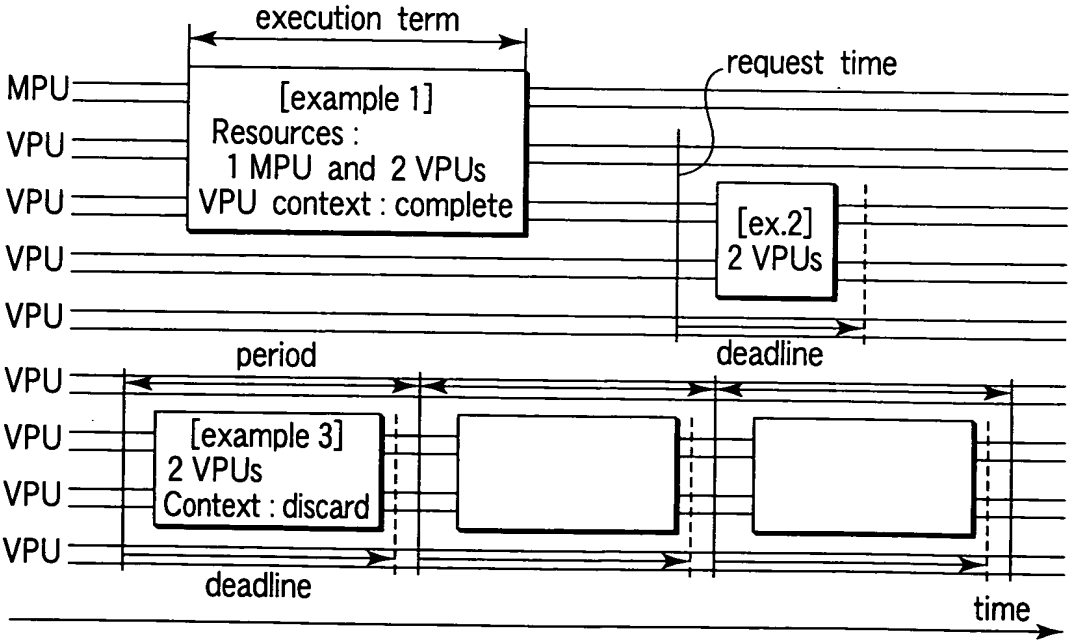


FIG. 55

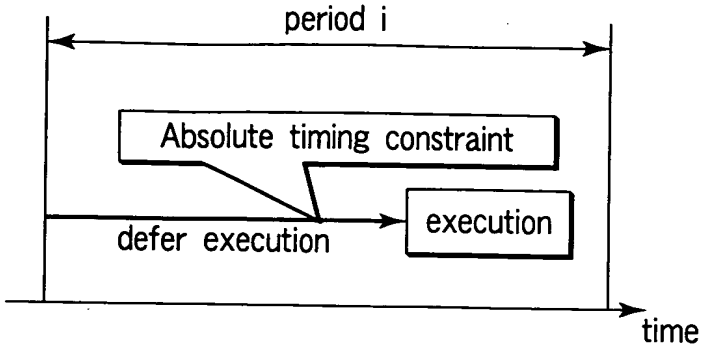


FIG. 56

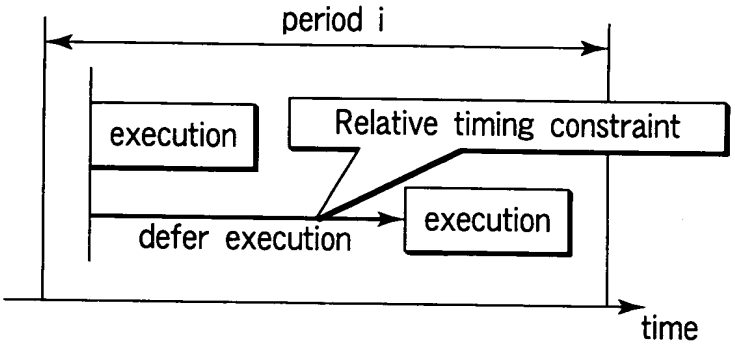


FIG. 57

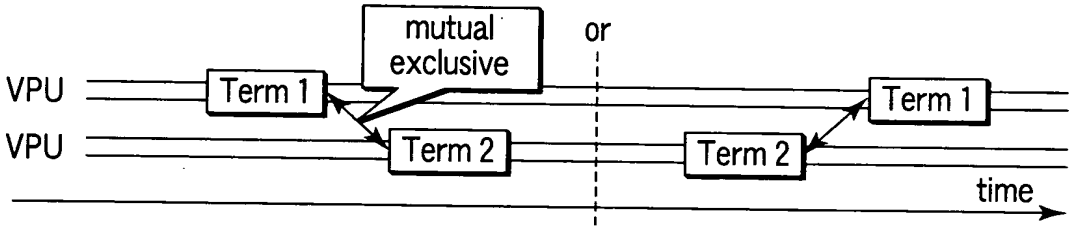


FIG. 58

| | | Tightly coupled thread group | Loosely coupled thread group |
|---|---|---|---|
| On memory | LS | Can use | Cannot use |
| | MS | Can use | |
| Other | | Should use hardware primitives | Should use mechanisms provided by VPU Runtime Environment |

FIG. 59

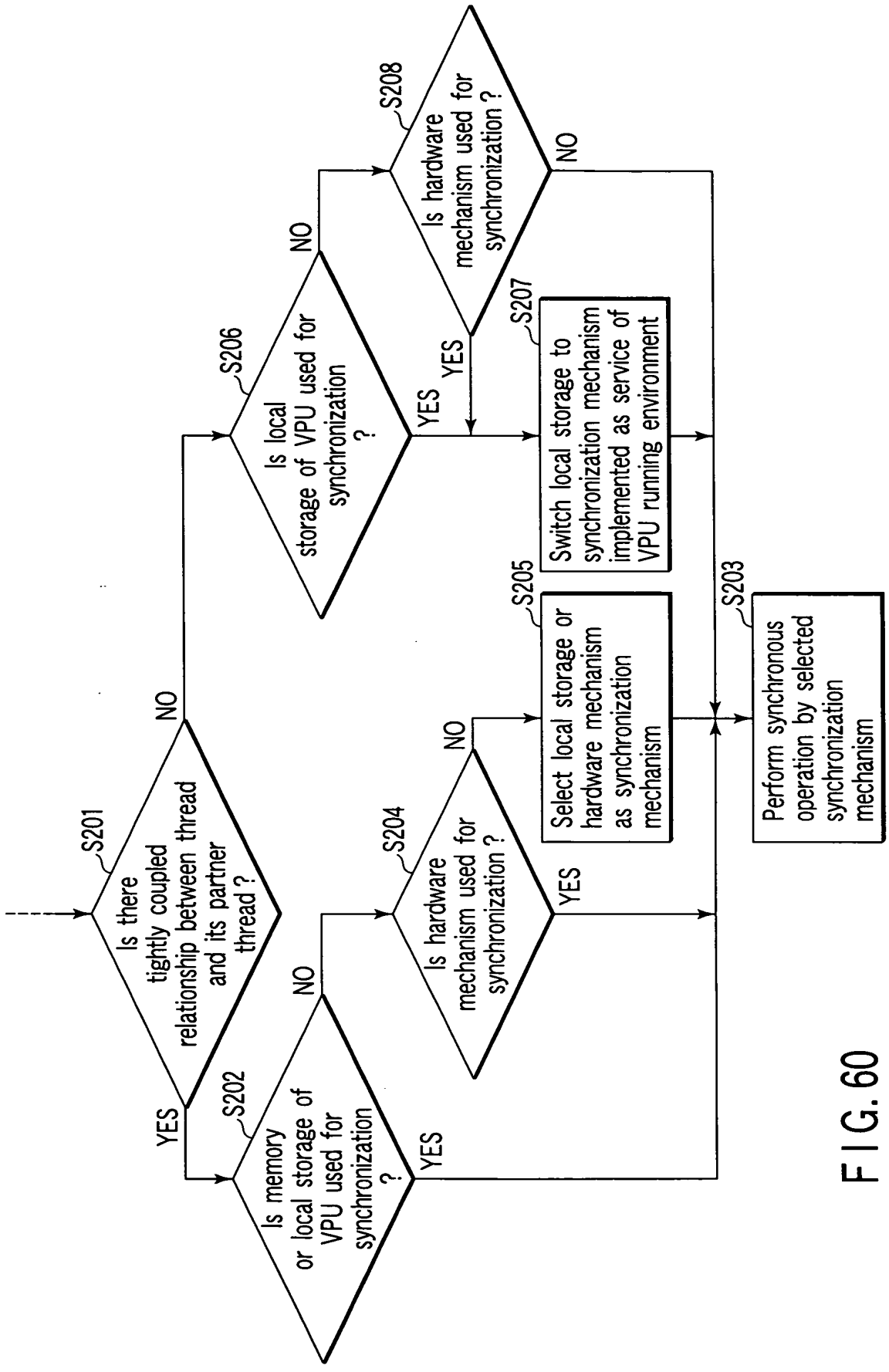
S201
Is there tightly coupled relationship between thread and its partner thread ?
YES
NO
S202
Is memory or local storage of VPU used for synchronization ?
YES
NO
S204
Is hardware mechanism used for synchronization ?
YES
NO
S205
Select local storage or hardware mechanism as synchronization mechanism
S206
Is local storage of VPU used for synchronization ?
YES
NO
S207
Switch local storage to synchronization mechanism implemented as service of VPU running environment
S208
Is hardware mechanism used for synchronization ?
YES
NO
S203
Perform synchronous operation by selected synchronization mechanism

FIG. 60

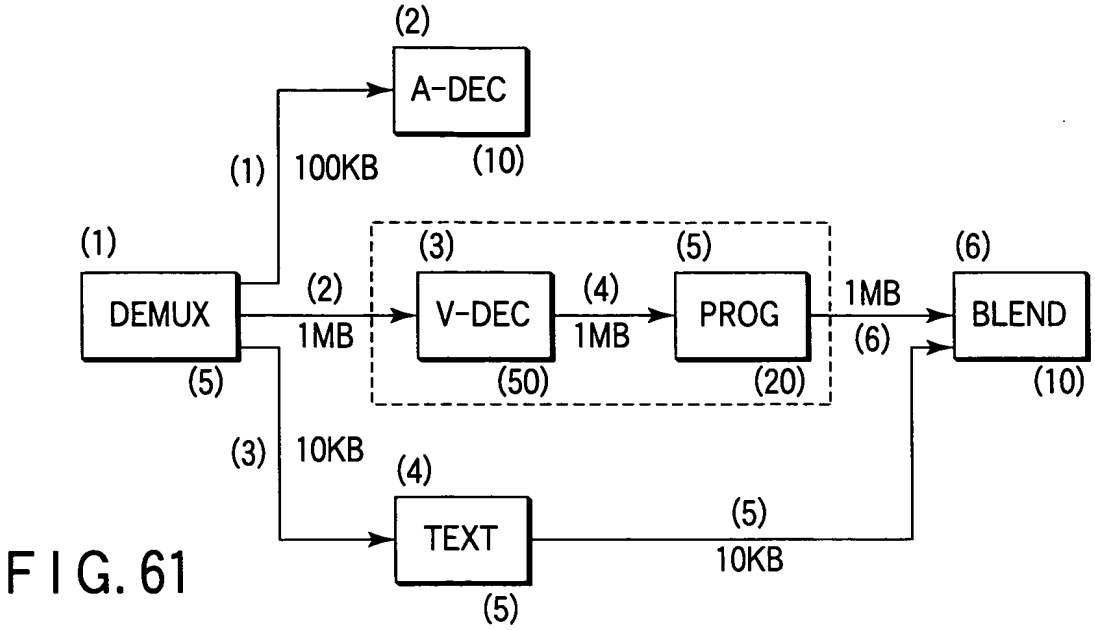


FIG. 61

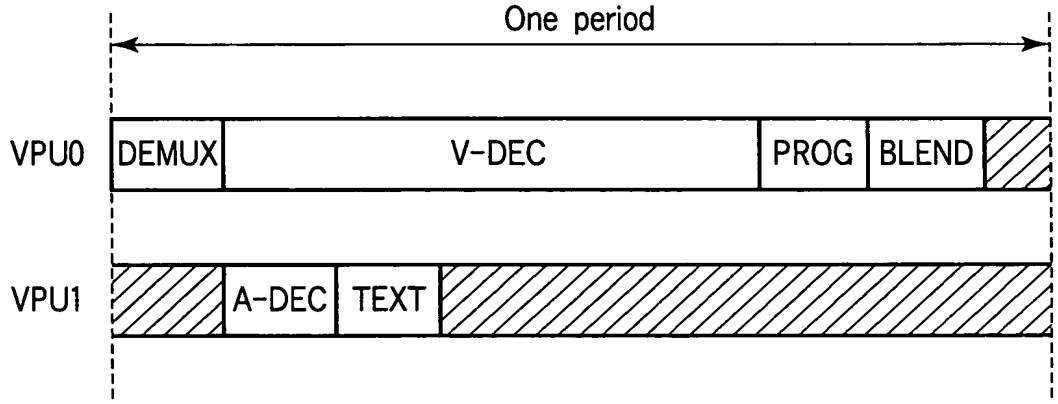


FIG. 63

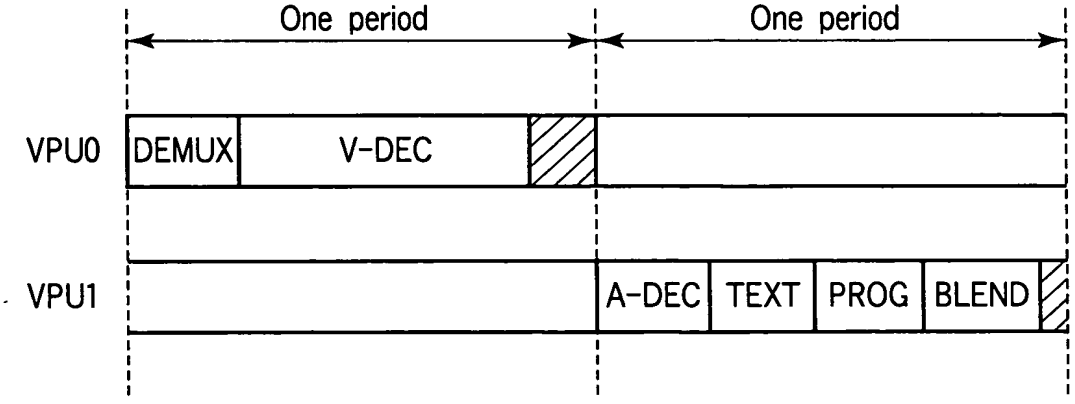


FIG. 64

| |
|---|
| BUFFER Id : 1<br>Size : 100KB SrcTask : 1 DstTask : 2 |
| BUFFER Id : 2<br>Size : 1MB SrcTask : 1 DstTask : 3 |
| BUFFER Id : 3<br>Size : 10KB SrcTask : 1 DstTask : 4 |
| BUFFER Id : 4<br>Size : 1MB SrcTask : 3 DstTask : 5 |
| BUFFER Id : 5<br>Size : 10KB SrcTask : 4 DstTask : 6 |
| BUFFER Id : 6<br>Size : 1MB SrcTask : 5 DstTask : 6 |
| TASK Id : 1 Class : VPU,HRT<br>ThreadContext : DEMUX Cost : 5<br>Constraint : Precede : 2,3,4<br>InputBuffer : OutputBuffer : 1,2,3 |
| TASK Id : 2 Class : VPU,HRT<br>ThreadContext : A-DEC Cost : 10<br>Constraint : Precede :<br>InputBuffer : 1 OutputBuffer : |
| TASK Id : 3 Class : VPU,HRT<br>ThreadContext : V-DEC Cost : 50<br>Constraint : Precede : 5<br>InputBuffer : 2 OutputBuffer : 4 |
| TASK Id : 4 Class : VPU,HRT<br>ThreadContext : TEXT Cost : 5<br>Constraint : Precede : 6<br>InputBuffer : 3 OutputBuffer : 5 |
| TASK Id : 5 Class : VPU,HRT<br>ThreadContext : PROG Cost : 20<br>Constraint : Precede : 6<br>InputBuffer : 4 OutputBuffer : 6 |
| TASK Id : 6 Class : VPU,HRT<br>ThreadContext : BLEND Cost : 10<br>Constraint : Precede : 5<br>InputBuffer : 5,6 OutputBuffer : |

FIG. 62

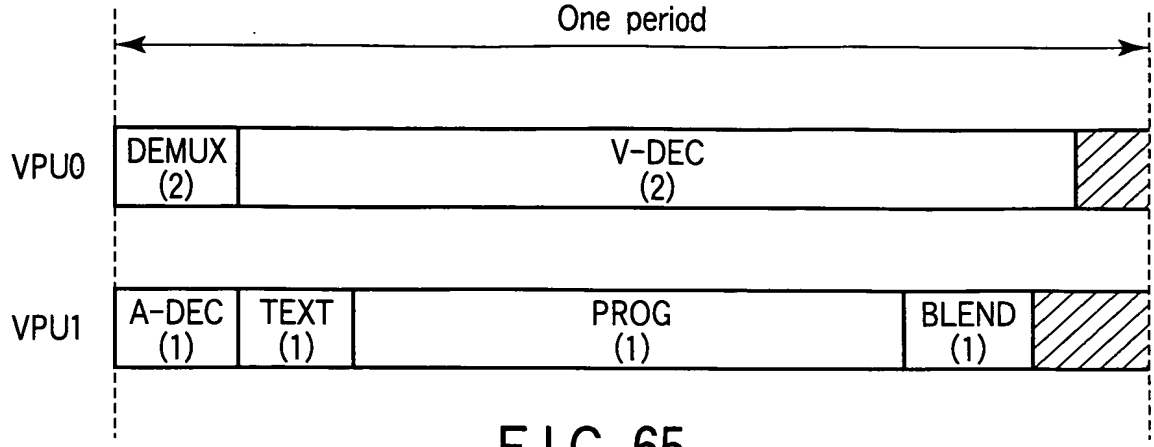


FIG. 65

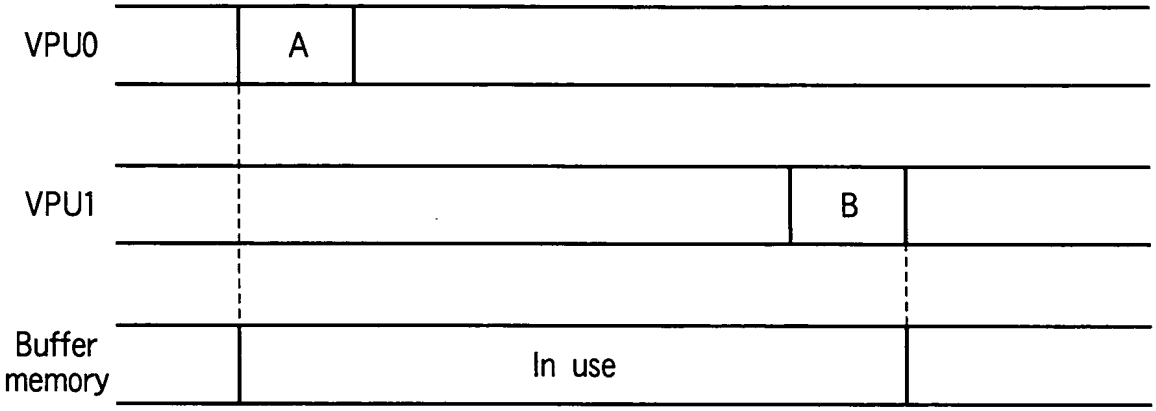


FIG. 66

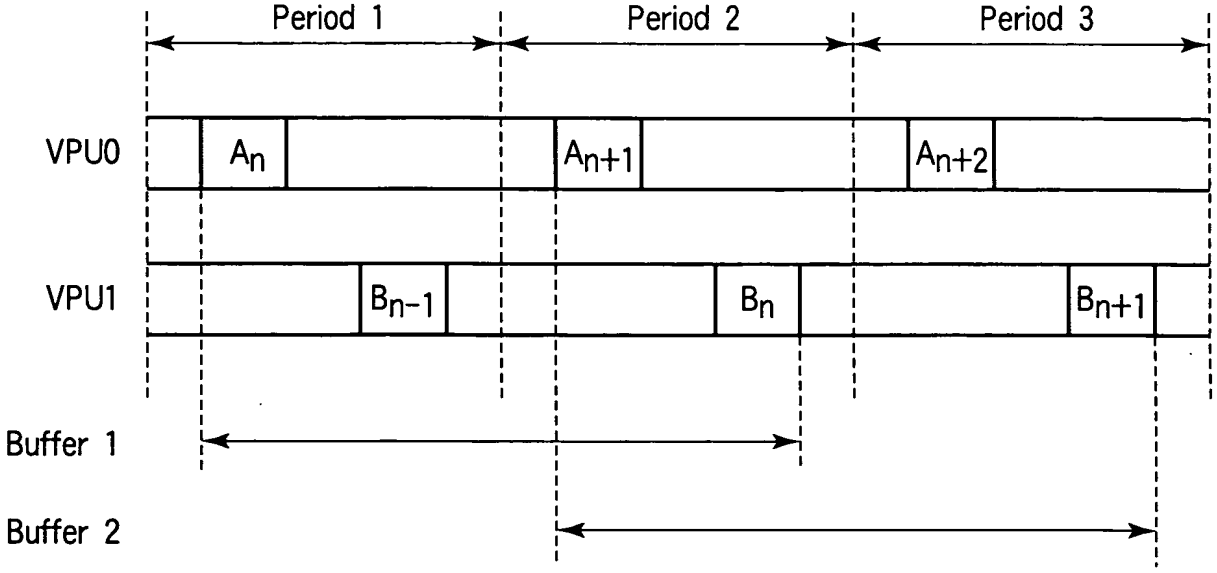


FIG. 67

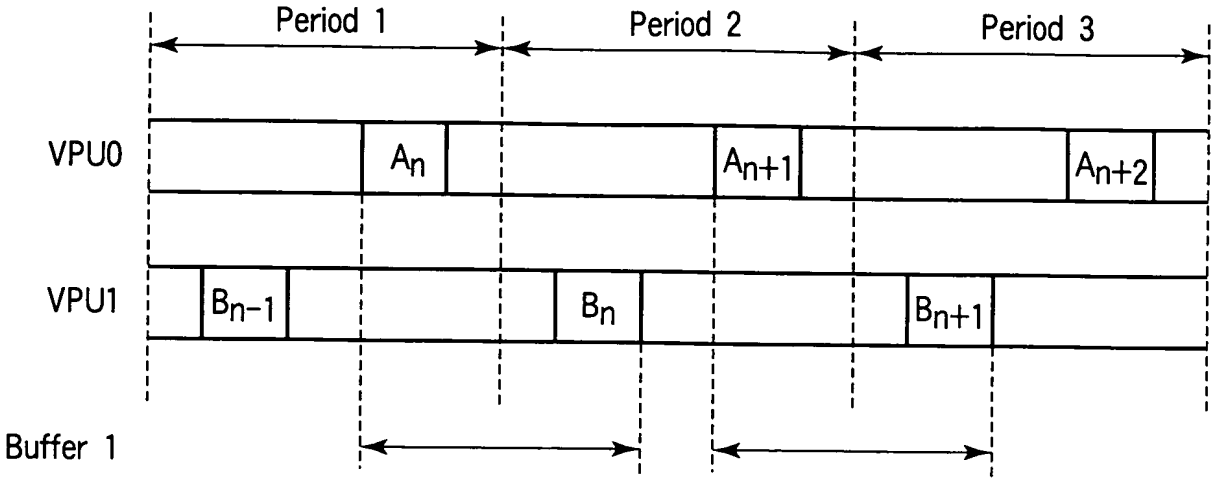


FIG. 68

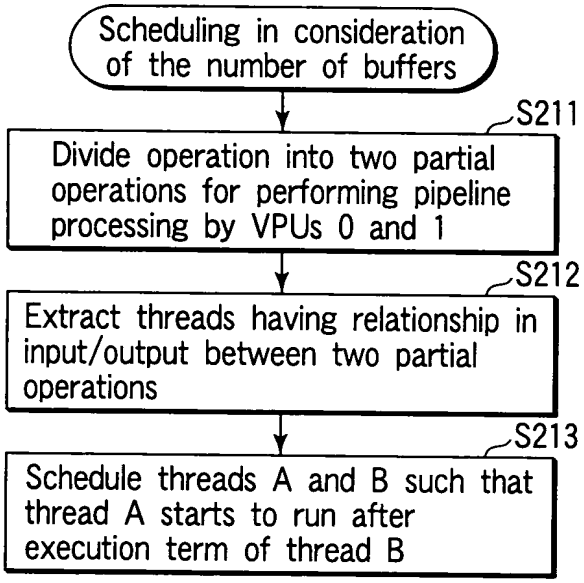


FIG. 69

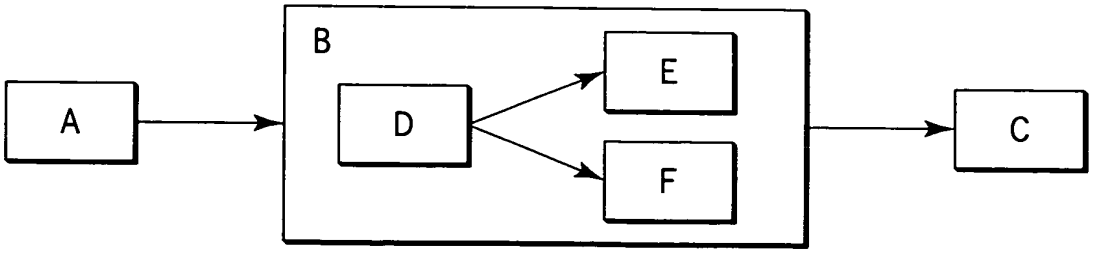


FIG. 70

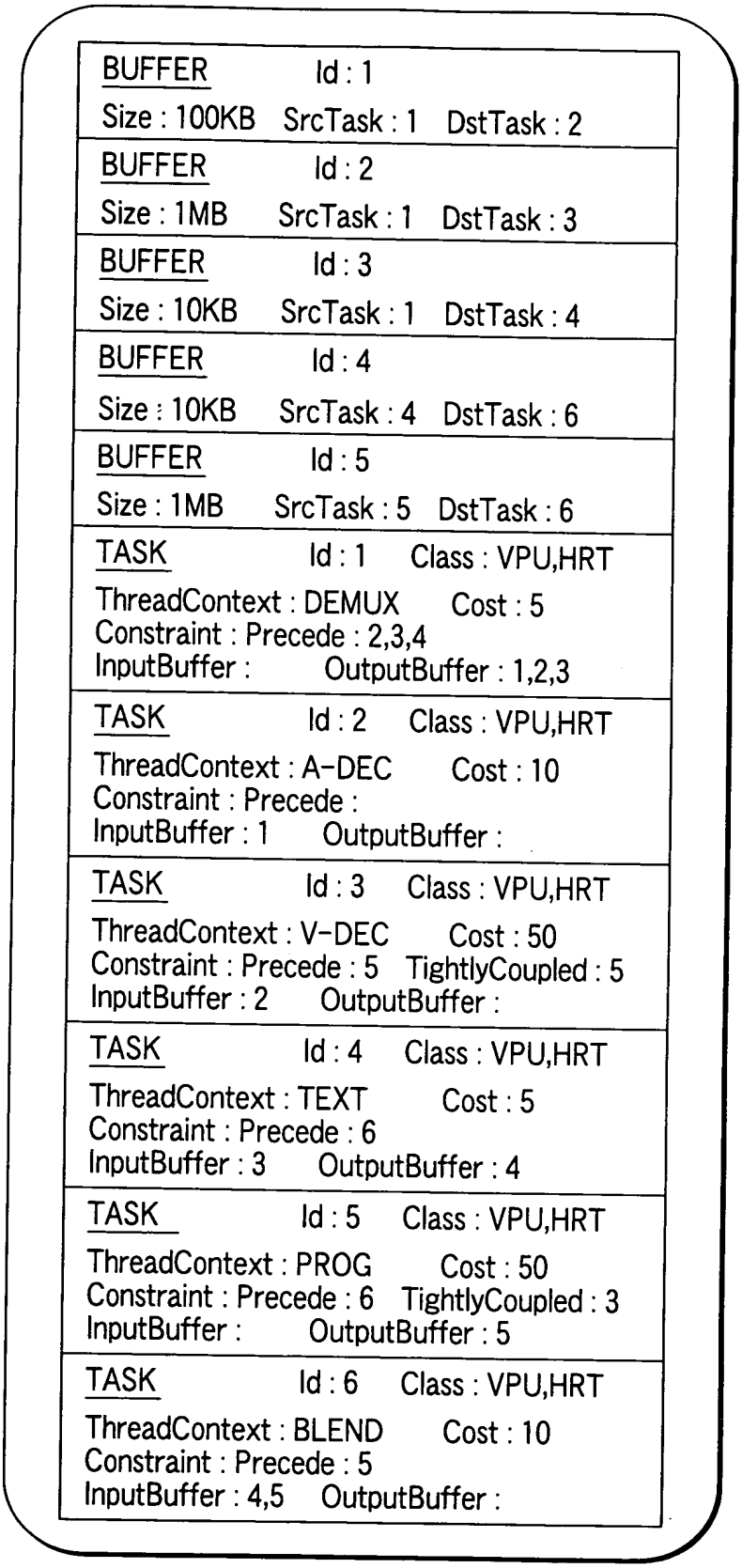
BUFFER Id : 1
Size : 100KB SrcTask : 1 DstTask : 2
BUFFER Id : 2
Size : 1MB SrcTask : 1 DstTask : 3
BUFFER Id : 3
Size : 10KB SrcTask : 1 DstTask : 4
BUFFER Id : 4
Size : 10KB SrcTask : 4 DstTask : 6
BUFFER Id : 5
Size : 1MB SrcTask : 5 DstTask : 6
TASK Id : 1 Class : VPU,HRT
ThreadContext : DEMUX Cost : 5
Constraint : Precede : 2,3,4
InputBuffer : OutputBuffer : 1,2,3
TASK Id : 2 Class : VPU,HRT
ThreadContext : A-DEC Cost : 10
Constraint : Precede :
InputBuffer : 1 OutputBuffer :
TASK Id : 3 Class : VPU,HRT
ThreadContext : V-DEC Cost : 50
Constraint : Precede : 5 TightlyCoupled : 5
InputBuffer : 2 OutputBuffer :
TASK Id : 4 Class : VPU,HRT
ThreadContext : TEXT Cost : 5
Constraint : Precede : 6
InputBuffer : 3 OutputBuffer : 4
TASK Id : 5 Class : VPU,HRT
ThreadContext : PROG Cost : 50
Constraint : Precede : 6 TightlyCoupled : 3
InputBuffer : OutputBuffer : 5
TASK Id : 6 Class : VPU,HRT
ThreadContext : BLEND Cost : 10
Constraint : Precede : 5
InputBuffer : 4,5 OutputBuffer :

FIG. 71

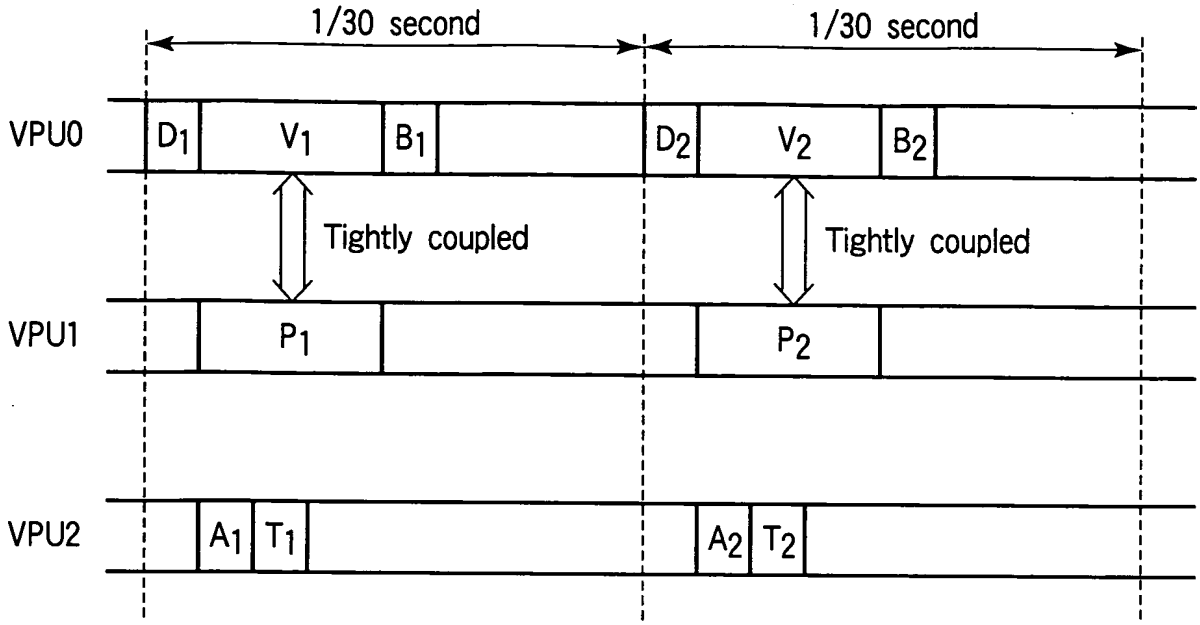
1/30 second
1/30 second
VPU0
D1
V1
B1
D2
V2
B2
Tightly coupled
Tightly coupled
VPU1
P1
P2
VPU2
A1
T1
A2
T2

FIG. 72

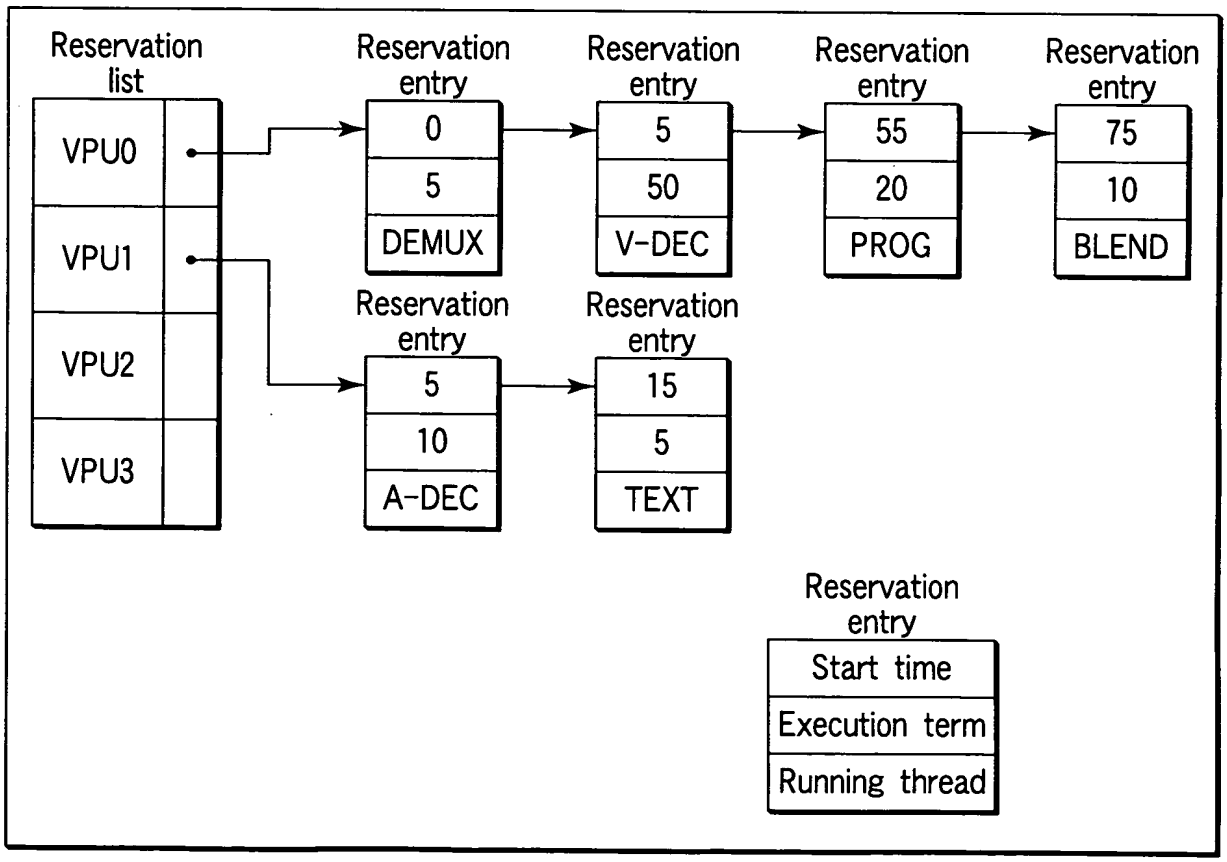
Reservation list
VPU0
VPU1
VPU2
VPU3
Reservation entry
0
5
DEMUX
Reservation entry
5
50
V-DEC
Reservation entry
55
20
PROG
Reservation entry
75
10
BLEND
Reservation entry
5
10
A-DEC
Reservation entry
15
5
TEXT
Reservation entry
Start time
Execution term
Running thread

FIG. 73

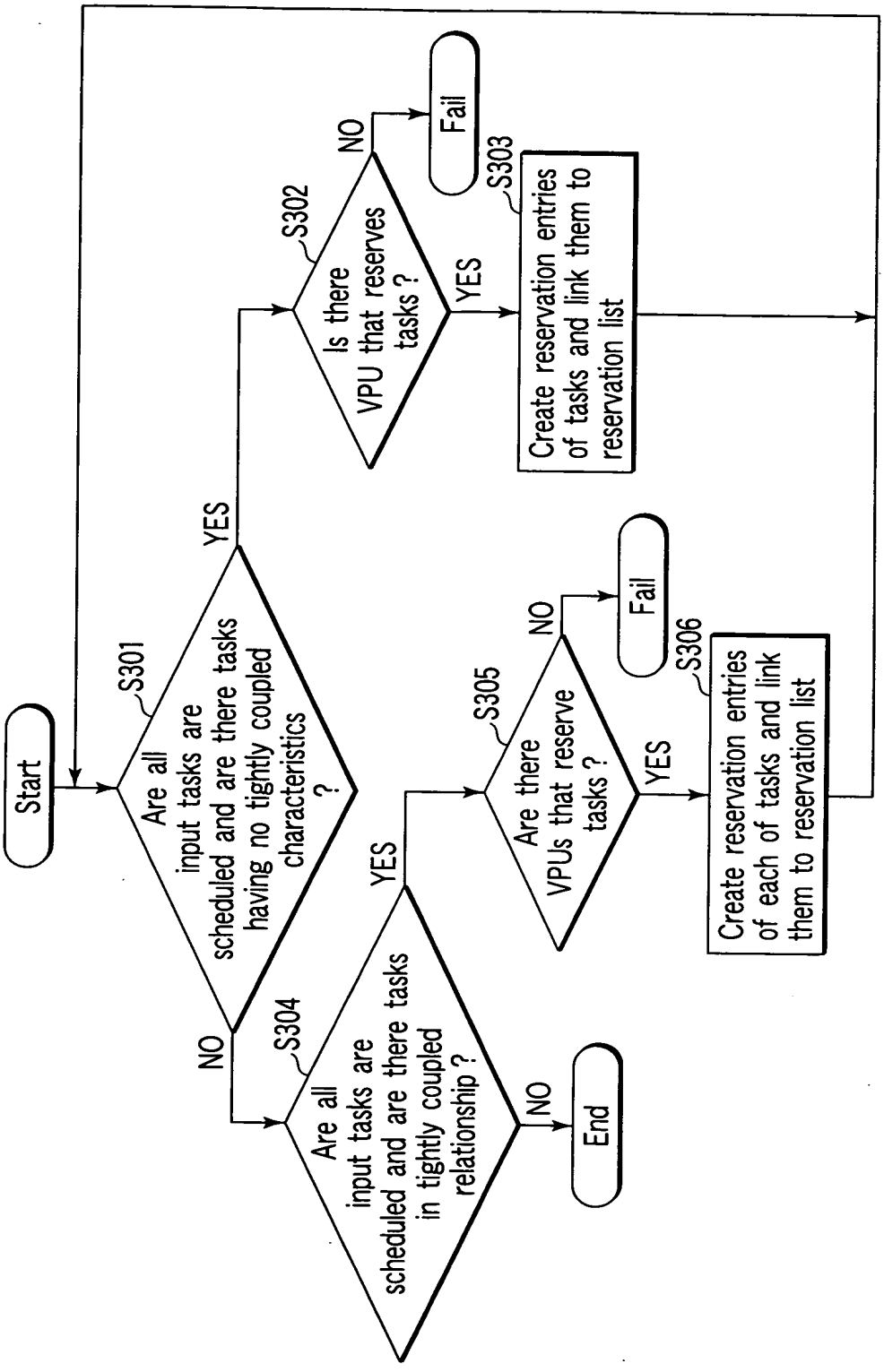
Start
S301
Are all input tasks are scheduled and are there tasks having no tightly coupled characteristics ?
YES
NO
S302
Is there VPU that reserves tasks ?
NO
Fail
YES
S303
Create reservation entries of tasks and link them to reservation list
S304
Are all input tasks are scheduled and are there tasks in tightly coupled relationship ?
YES
NO
End
S305
Are there VPUs that reserve tasks ?
NO
Fail
YES
S306
Create reservation entries of each of tasks and link them to reservation list

FIG. 74